

**SEQUENTIAL PATTERN MINING PADA PENCARIAN POLA
PERILAKU PENGGUNA INTERNET MENGGUNAKAN
ALGORITMA SPADE**

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun oleh :

DIKA RIZKY YUNianto
NIM. 105060800111031

**PROGRAM STUDI INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2013**

**LEMBAR PERSETUJUAN
SEQUENTIAL PATTERN MINING PADA PENCARIAN POLA
PERILAKU PENGGUNA INTERNET MENGGUNAKAN
ALGORITMA SPADE**

SKRIPSI



Disusun oleh :
DIKA RIZKY YUNianto
NIM. 105060800111031

Telah diperiksa dan disetujui oleh :

Pembimbing I,

Pembimbing II,

Candra Dewi, S.Kom.,M.Sc.
NIP. 19771114 200312 2 001

Novanto Yudistira, S.Kom.,M.Sc.
NIK. 83111016110425

LEMBAR PENGESAHAN
SEQUENTIAL PATTERN MINING PADA PENCARIAN POLA
PERILAKU PENGGUNA INTERNET MENGGUNAKAN
ALGORITMA SPADE

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh :

DIKA RIZKY YUNianto

NIM. 105060800111031

Skripsi ini telah diuji dan dinyatakan lulus tanggal 9 Oktober 2014

Penguji I

Penguji II

Indriati, ST., M.Kom.
NIK. 831013 06 1 2 0035

Satrio Agung Wicaksono, S.Kom., M.Kom.
NIP. 19860521 201212 1 001

Penguji III

Imam Cholissodin, S.Si., M.Kom.
NIK. 850719 16 1 1 0422

Mengetahui

Ketua Program Studi Teknik Informatika / Ilmu Komputer

Drs. Marji, MT.
NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).



Malang, Oktober 2014

Mahasiswa,

Dika Rizky Yunianto

NIM 105060800111031

KATA PENGANTAR

Alhamdulillahirobbilalamin, puji dan syukur kehadiran Allah *Subhanahu Wa Ta'ala*, karena atas segala rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi yang berjudul ***Sequential Pattern Mining Pada Pencarian Pola Perilaku Penggunaan Internet Menggunakan Algoritma SPADE***. Tak lupa, shalawat serta salam penulis haturkan kepada junjungan besar, Nabi Muhammad *Shalallahu Alaihi Wasallam*.

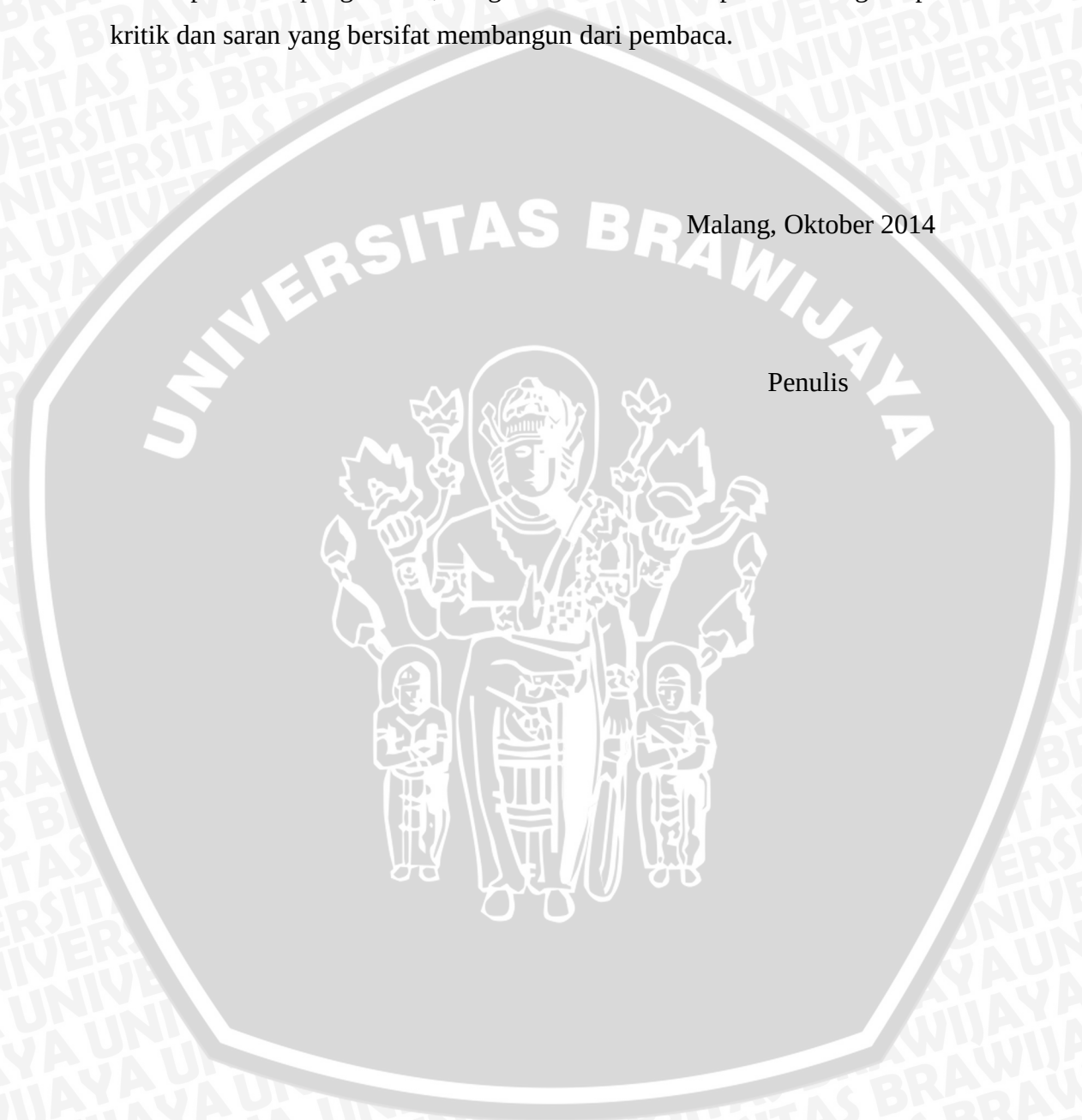
Penyusunan skripsi ini juga tak lepas dari bantuan baik itu berupa bimbingan, kritik, saran, dukungan, motivasi maupun doa banyak pihak. Oleh karena itu, ucapan terima kasih penulis sampaikan kepada :

1. Kedua orang tua penulis yang tersayang dan tercinta atas segala dukungan, motivasi, doa dan segala pengorbanan baik materil maupun moril.
2. Candra Dewi, S.Kom.,M.Sc. selaku dosen pembimbing pertama dan Novanto Yudistira, S.Kom.,M.Sc. selaku dosen pembimbing kedua atas segala bimbingan, waktu yang diluangkan serta kritik dan saran yang telah diberikan kepada penulis.
3. Seluruh bapak dan ibu dosen Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang atas segala bimbingan serta ilmu yang telah diajarkan kepada penulis.
4. Para pegawai dan staf Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya Malang.
5. Mas Alan dan seluruh staff PPTI-UB atas bantuan dan waktu yang telah diberikan.
6. Kakak tercinta dan tersayang, Diah Ivana Sari atas dukungan dan motivasinya serta keponakan tersayang, Naufal Fayzan Almadina.
7. Rizal Bagus, Dimas Widiastara, Hariz Farisi, Citra Krisandra, Vashty Armila atas segala bantuan, motivasi dan semangat yang telah diberikan.
8. Teman-teman seperjuangan (Shindu, Dhinda, Depru, Rinaldy, Geby, Irwan, Ayun, Arum).
9. Dan semua pihak yang tidak bisa disebutkan satu per satu. Terima kasih atas segala bantuannya.

Semoga penulisan laporan skripsi ini bermanfaat bagi pembaca dan untuk pengembangan selanjutnya. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna serta banyak kekurangan disebabkan oleh keterbatasan kemampuan dan pengalaman, dengan kerendahan hati penulis mengharapkan kritik dan saran yang bersifat membangun dari pembaca.

Malang, Oktober 2014

Penulis



ABSTRAK

Dika Rizky Yunianto. 2014. *Sequential Pattern Mining* Pada Pencarian Pola Perilaku Penggunaan *Internet* Menggunakan Algoritma SPADE.

Dosen Pembimbing: Candra Dewi, S.Kom.,M.Sc. dan Novanto Yudistira, S.Kom.,M.Sc.

Perilaku penggunaan *internet* dapat dilihat melalui sebuah file yang mencatat segala informasi transaksi penggunaan *internet* yang disebut *access logs*. Algoritma *sequential pattern discovery using equivalent classes* (SPADE) merupakan salah satu algoritma dari *sequential pattern mining* yang digunakan untuk menemukan pola agar mendapatkan informasi yang berguna. Nilai minimum *support* dan minimum *confidence* ditentukan untuk menemukan *rule-rule* yang bermanfaat. Nilai kekuatan *rule* atau *lift ratio* ditentukan dengan besar minimum *support* sedangkan nilai minimum *confidence* tidak berpengaruh terhadap besar nilai *lift ratio*. Algoritma SPADE (*sequential pattern discovery using equivalent classes*) diterapkan untuk mencari pola perilaku pengguna *internet* dengan cara melakukan *preprocessing* data untuk menyaring informasi yang dibutuhkan. Dilanjutkan dengan pembentukan data transaksi dan perhitungan SPADE dengan mengkombinasikan *itemset* dan menghitung *frequent*-nya untuk mendapatkan *rule* yang kemudian akan dicari kekuatan setiap *rule* dengan menghitung *lift ratio*-nya. *Rule* yang kuat adalah *rule* yang memiliki nilai *lift ratio* lebih besar dari 1. Tingkat kekuatan atau *lift ratio* dari *rule* yang dihasilkan pada tiap fakultas berbeda-beda. Nilai *lift ratio* terbesar terdapat pada Fakultas Peternakan disaat jam kerja dengan besaran nilai 18,868 dengan *rule* jika mengakses mozilla.com maka mengakses bbc.co.uk.

Kata kunci: Sequential Pattern Mining, Pencarian Pola, SPADE

ABSTRACT

Dika Rizky Yunianto. 2014. *Sequential Pattern Mining for Search of Internet User Behavior Pattern Using SPADE Algorithm.*

Advisor: Candra Dewi, S.Kom.,M.Sc. and Novanto Yudistira, S.Kom.,M.Sc.

Internet usage behavior can be seen through a file that records all transaction information using the Internet called access logs. The algorithm sequential pattern discovery using equivalent classes (SPADE) is one of the sequential pattern mining algorithm is used to find patterns in order to obtain useful information. Value of minimum support and minimum confidence is determined to find some rules as helpful. Value of lift ratio is determined by the value of the minimum support while the minimum confidence value has no effect on the lift ratio value. Algorithm SPADE (Sequential pattern discovery using Equivalent classes) is applied to search for patterns of behavior of Internet users by way of preprocessing the data to filter out the needed information. Followed by the establishment of transaction data and calculations by combining SPADE frequent itemset and calculate its to get a rule which will then look for the strength of each rule by calculating the lift ratio. Strong rule is a rule that has a lift ratio greater than 1. Strength ratio of each rule generated at different faculties. Value elevator largest ratio found in the Faculty of Husbandry when amount of work hours with value 18.868 , if access mozilla.com then accessing bbc.co.uk..

Keywords: Sequential Pattern Mining, Search Pattern, SPADE

DAFTAR ISI

LEMBAR PERSETUJUAN	i
LEMBAR PENGESAHAN.....	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRAC	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR SOURCECODE.....	xiv
DAFTAR PERSAMAAN	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	6
2.1 Penelitian Terkait	6
2.2 Pola Perilaku.....	6
2.3 Internet	6
2.4 Website	7
2.5 URL (Universal Resource Locator)	7
2.6 DNS (Domain Name Space)	8
2.7 IP Address	9
2.8 Access Logs	10
2.9 Data Mining	11
2.10 Association Rule	12

2.11	Support	13
2.12	Confidence	13
2.13	Sequential Pattern Mining	13
2.14	SPADE (Sequential Pattern Discovery using Ecuivalent Classes) ..	14
2.15	Lift Ratio	17
BAB III METODOLOGI DAN PERANCANGAN SISTEM		18
3.1	Metodologi	18
3.2	Pengumpulan Data	19
3.3	Analisis dan Perancangan Sistem	19
3.3.1	Deskripsi Sistem	19
3.3.2	Batasan Sistem	22
3.4	Perancangan Proses	22
3.4.1	Preprocessing Data Access Logs	22
3.4.2	Pembentukan Data Transaksi	26
3.4.3	Perhitungan SPADE	27
3.4.3.1	Pembentukan 1-Sequence	29
3.4.3.2	Pembentukan 2-Sequence	30
3.4.3.3	Pembentukan k-Sequence	32
3.4.4	Perhitungan Lift Ratio	34
3.5	Perhitungan Manual	35
3.6	Perancangan Antarmuka (User Interface)	44
3.7	Perancangan Pengujian	48
BAB IV IMPLEMENTASI		49
4.1	Lingkungan Implementasi	49
4.1.1	Lingkunan Perangkat Keras	49
4.1.2	Lingkungan Perangkat Lunak	49
4.2	Implementasi Program.....	50
4.2.1	Implementasi <i>Preprocessing Data Access Logs</i>	50
4.2.2	Implementasi Pembentukan Data Transaksi.....	51
4.2.3	Implementasi Pembentukan SPADE.....	52
4.2.3.1	<i>Frequent 1-Sequence</i>	53
4.2.3.2	<i>Frequent 2-Sequence</i>	54

4.2.3.3	<i>Frequent k-Sequence</i>	55
4.2.4	Implementasi Perhitungan <i>Lift Ratio</i>	57
4.3	Implementasi Antarmuka	58
BAB V PENGUJIAN DAN ANALISIS		61
5.1	Hasil Pengujian	61
5.1.1	Hasil Pengujian Berdasarkan Minimum <i>Support</i>	62
5.1.2	Hasil Pengujian Berdasarkan Minimum <i>Confidence</i>	65
5.1.3	Hasil Pengujian Akurasi Algoritma SPADE	69
5.1.4	Hasil Pengujian Rule Terbaik	69
5.1.4.1	Fakultas Teknik	69
5.1.4.2	Fakultas Matematika dan Ilmu Pengetahuan	70
5.1.4.3	Fakultas Teknologi Pertanian	70
5.1.4.4	Fakultas Kedokteran	71
5.1.4.5	Fakultas Pertanian	71
5.1.4.6	Fakultas Peternakan	71
5.1.4.7	Fakultas Perikanan dan Ilmu Kelautan	72
5.2	Analisa	72
BAB IV PENUTUP		74
6.1	Kesimpulan	74
6.2	Saran	75
DAFTAR PUSTAKA		76
LAMPIRAN A		79
LAMPIRAN B		81
LAMPIRAN C		80

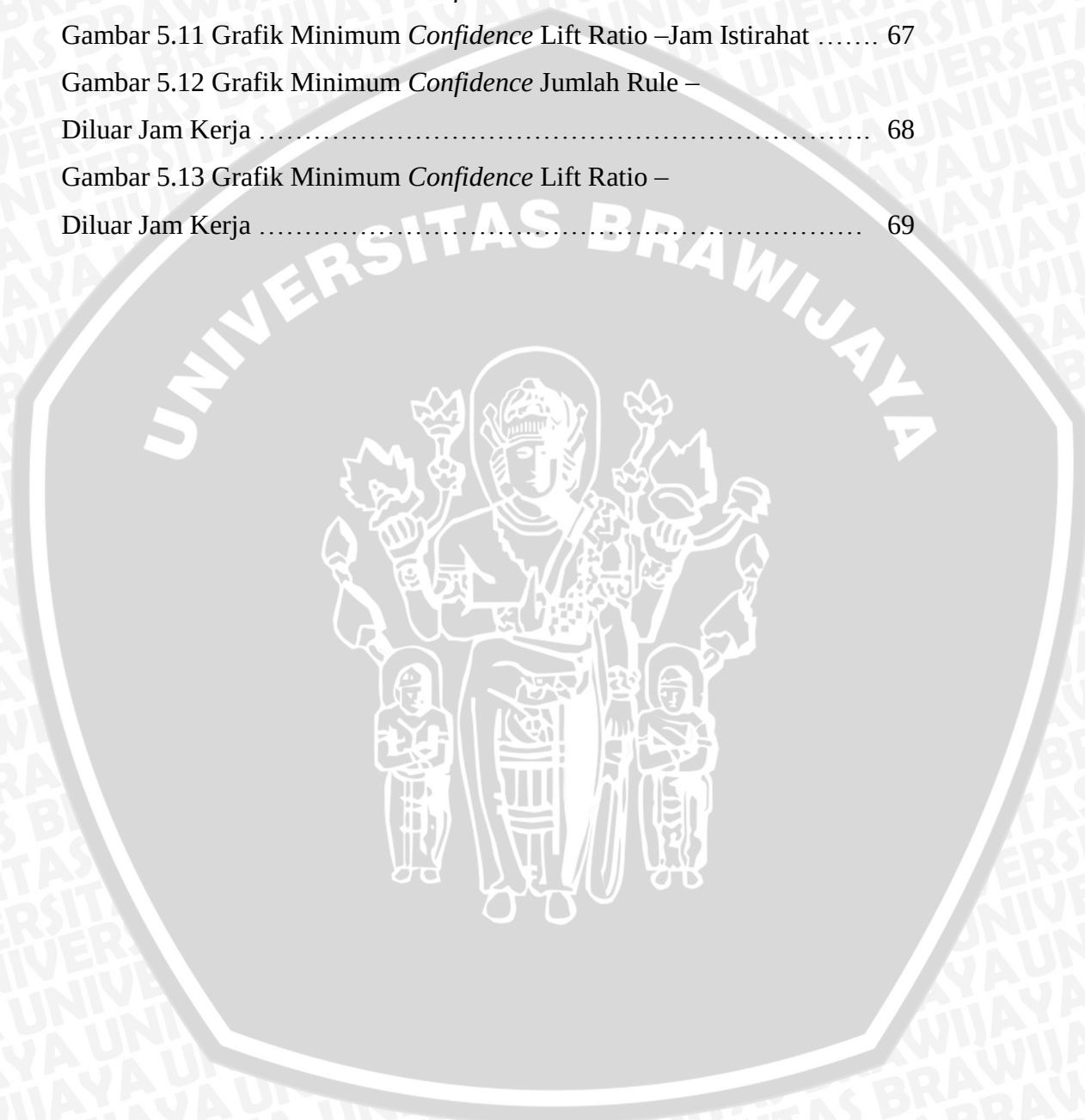
DAFTAR TABEL

Tabel 2.1 Bentuk umum data <i>access logs</i>	10
Tabel 2.2 Contoh data transaksi	19
Tabel 3.1. Contoh data <i>access logs</i> pada <i>proxy</i> di Universitas Brawijaya...	20
Tabel 3.2. Nama-nama fakultas ilmu eksak di Universitas Brawijaya.....	20
Tabel 3.3. Pembagian kategori waktu	22
Tabel 3.4. Alokasi <i>IP address</i> Fakultas-fakultas Ilmu Eksak di Universitas Brawijaya	22
Tabel 3.5. Elemen-elemen pada <i>Data Access Logs</i>	23
Tabel 3.6. Hasil <i>Filtering</i> data <i>Access Logs</i>	24
Tabel 3.7. Contoh data <i>access logs</i> pada <i>proxy</i> Universitas Brawijaya	36
Tabel 3.8. Contoh pengelompokan IP dan penamaan SID serta EID	37
Tabel 3.9. Data transaksi <i>access logs</i>	38
Tabel 3.10. Indeks <i>website</i>	39
Tabel 3.11. Data transaksi <i>sequence</i> vertikal	40
Tabel 3.12. <i>Frequent 1-sequence</i>	41
Tabel 3.13. <i>Frequent 2-sequence</i>	42
Tabel 3.14. <i>Frequent 3-sequence</i>	43
Tabel 3.15. <i>Lift ratio rule</i>	43
Tabel 5.1 Tabel <i>rule</i> terbaik Fakultas Teknik	70
Tabel 5.2 Tabel <i>rule</i> terbaik Fakultas MIPA	70
Tabel 5.3 Tabel <i>rule</i> terbaik Fakultas Teknologi Pertanian	70
Tabel 5.4 Tabel <i>rule</i> terbaik Fakultas Kedokteran.....	71
Tabel 5.5 Tabel <i>rule</i> terbaik Fakultas Pertanian.....	71
Tabel 5.6 Tabel <i>rule</i> terbaik Fakultas Peternakan.....	71
Tabel 5.7 Tabel <i>rule</i> terbaik Fakultas Perikanan dan Ilmu Kelautan.....	72

DAFTAR GAMBAR

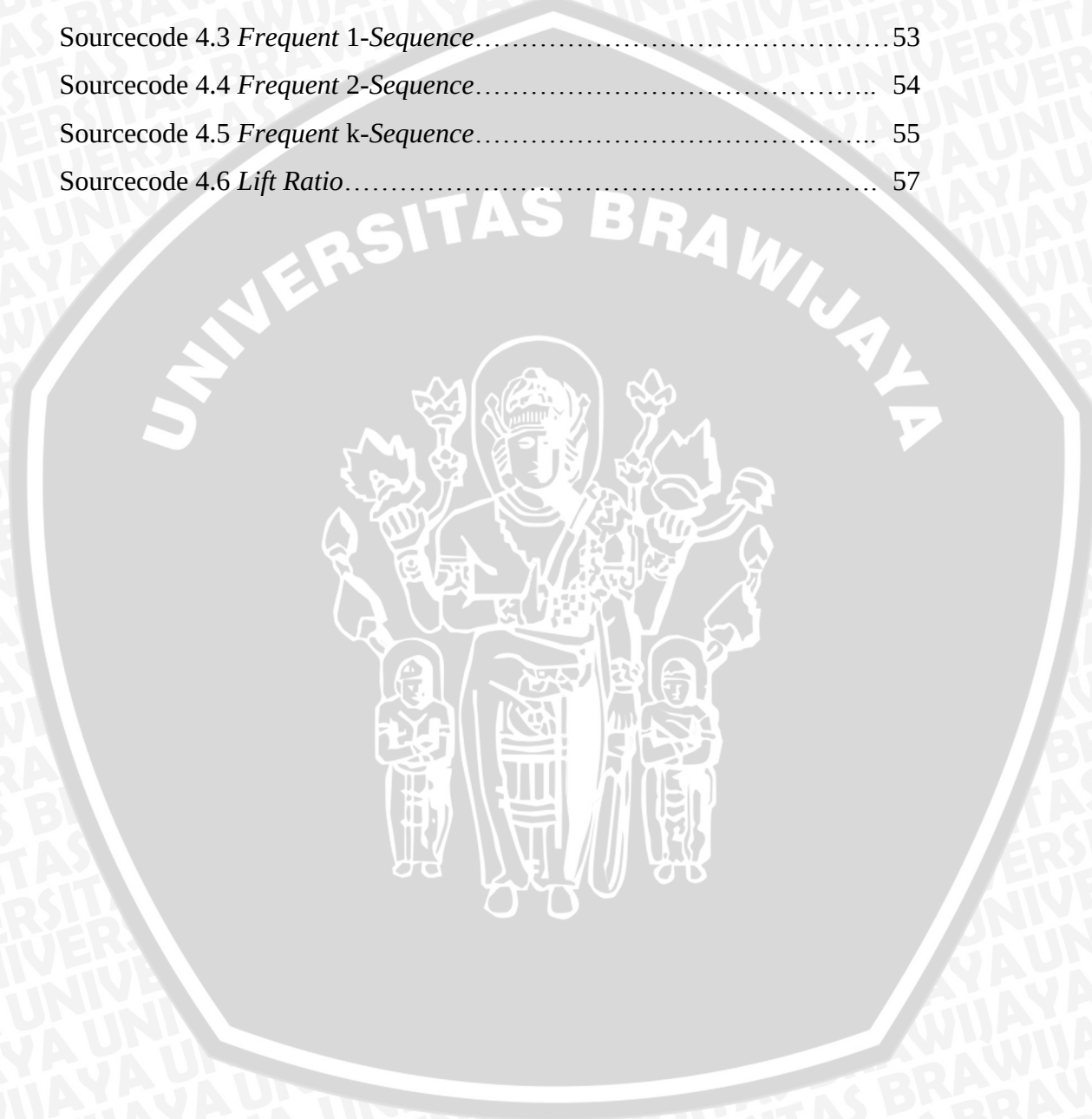
Gambar 2.1 Hirarki DNS Sumber: [ART-03]	9
Gambar 2.2 Proses <i>knowledge discovery in database</i> (KDD) Sumber [HAN-01].	11
Gambar 3.1. <i>Flowchart</i> langkah-langkah kerja penelitian.....	18
Gambar 3.2. Alur proses umum sistem	21
Gambar 3.3. <i>Flowchart</i> proses <i>preprocessing</i> data.....	25
Gambar 3.4. <i>Flowchart</i> proses pembentukan data transaksi.....	26
Gambar 3.5. <i>Flowchart</i> perhitungan SPADE	28
Gambar 3.6. <i>Flowchart</i> pembentukan <i>frequent 1-sequence</i>	29
Gambar 3.7. <i>Flowchart</i> pembentukan <i>frequent 2-sequence</i>	31
Gambar 3.8. <i>Flowchart</i> pembentukan frequent k-sequence.....	33
Gambar 3.9. <i>Flowchart</i> perhitungan lift ratio	34
Gambar 3.10. Rancangan <i>User Interface</i> halaman <i>preprocessing</i> data.....	44
Gambar 3.11. Rancangan <i>User Interface</i> halaman utama.....	45
Gambar 3.12. Rancangan <i>User Interface</i> halaman utama setelah data transaksi terbentuk.	46
Gambar 3.13. Rancangan <i>User Interface</i> halaman hasil perhitungan.....	47
Gambar 3.14. Rancangan <i>User Interface</i> halaman <i>index website</i>	48
Gambar 4.7. Halaman <i>preprocessing</i> data.....	58
Gambar 4.8. Halaman utama.....	59
Gambar 4.9. Halaman utama setelah didapatkan data transaksi.....	59
Gambar 4.10. Halaman hasil.....	60
Gambar 4.11. Halaman <i>index web</i>	60
Gambar 5.1 Grafik Jumlah Transaksi.....	61
Gambar 5.2 Grafik Minimum <i>Support</i> Jumlah Rule - Jam Kerja	62
Gambar 5.3 Grafik Minimum <i>Support</i> Lift Ratio- Jam Kerja	63
Gambar 5.4 Grafik Minimum <i>Support</i> Jumlah Rule - Jam Istirahat.....	63
Gambar 5.5 Grafik Minimum <i>Support</i> Lift Ratio- Jam Istirahat	64
Gambar 5.6 Grafik Minimum <i>Support</i> Jumlah Rule – Diluar Jam Kerja	64

Gambar 5.7 Grafik Minimum <i>Support Lift Ratio</i> - Diluar Jam Kerja	65
Gambar 5.8 Grafik Minimum <i>Confidence</i> Jumlah Rule –Jam Kerja	66
Gambar 5.9 Grafik Minimum <i>Confidence Lift Ratio</i> –Jam Kerja	66
Gambar 5.10 Grafik Minimum <i>Confidence</i> Jumlah Rule –Jam Istirahat	67
Gambar 5.11 Grafik Minimum <i>Confidence Lift Ratio</i> –Jam Istirahat	67
Gambar 5.12 Grafik Minimum <i>Confidence</i> Jumlah Rule – Diluar Jam Kerja	68
Gambar 5.13 Grafik Minimum <i>Confidence Lift Ratio</i> – Diluar Jam Kerja	69



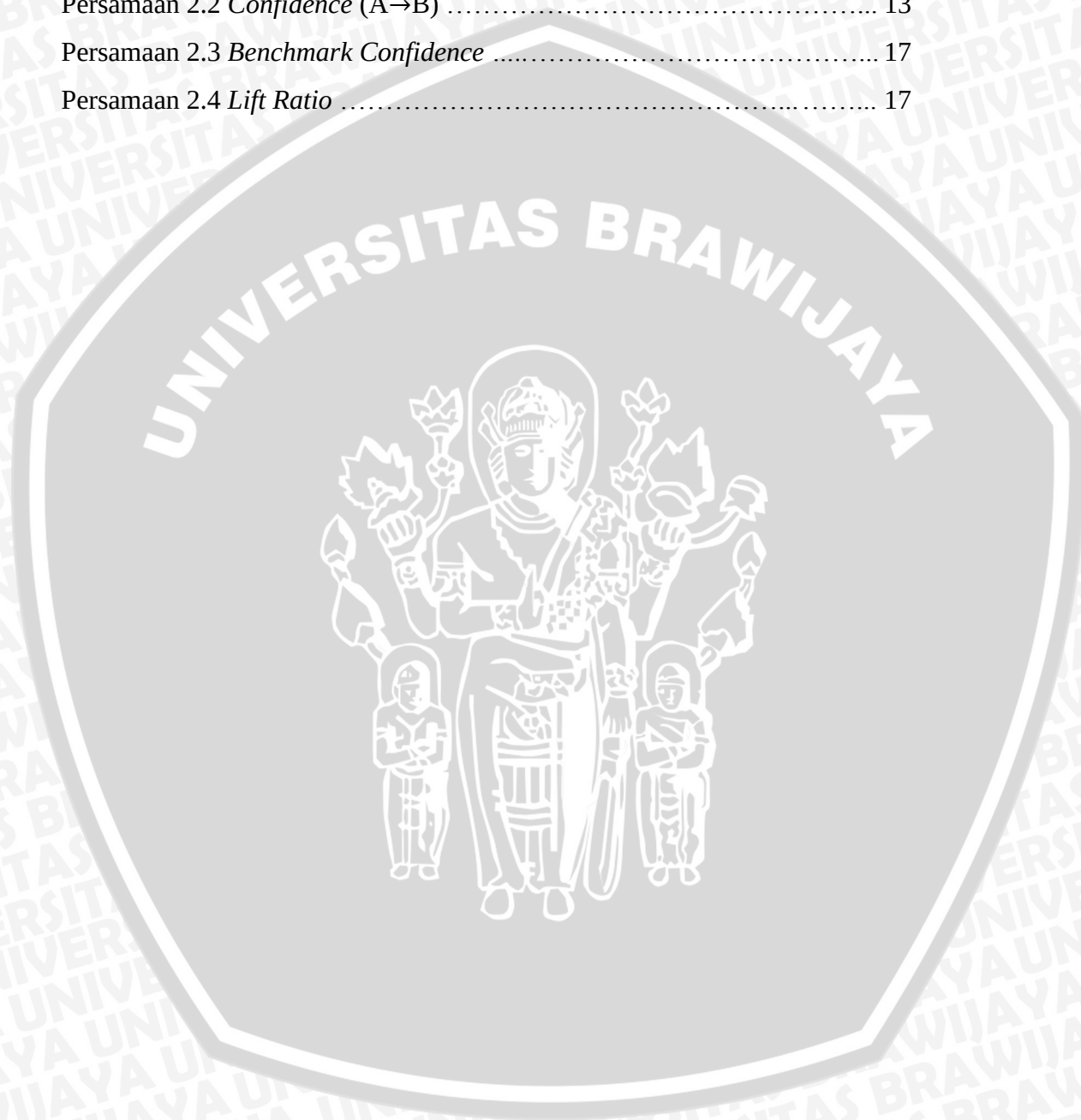
DAFTAR SOURCECODE

Sourcecode 4.1 <i>Preprocessing data Access Logs</i>	50
Sourcecode 4.2 <i>Pembentukan Data Transaksi</i>	51
Sourcecode 4.3 <i>Frequent 1-Sequence</i>	53
Sourcecode 4.4 <i>Frequent 2-Sequence</i>	54
Sourcecode 4.5 <i>Frequent k-Sequence</i>	55
Sourcecode 4.6 <i>Lift Ratio</i>	57



DAFTAR PERSAMAAN

Persamaan 2.1 <i>Support</i> ($A \rightarrow B$)	13
Persamaan 2.2 <i>Confidence</i> ($A \rightarrow B$)	13
Persamaan 2.3 <i>Benchmark Confidence</i>	17
Persamaan 2.4 <i>Lift Ratio</i>	17



BAB I

PENDAHULUAN

1.1 Latar Belakang

Universitas Brawijaya menyediakan fasilitas *internet* yang dapat diakses selama 24 jam oleh seluruh civitas akademika. Fasilitas *internet* ini digunakan sebagai penunjang kegiatan akademis, seperti pencarian literatur menggunakan *search engine*, pengumpulan tugas dan mengakses informasi-informasi terbaru mengenai kegiatan akademik selain itu, *internet* juga dapat digunakan sebagai media hiburan seperti mengakses *social media*, mendengarkan musik atau melihat video secara *online* serta berbelanja barang-barang kebutuhan melalui *online shop*. Peranan *internet* sangat penting untuk mendukung kegiatan ilmiah baik dikalangan ilmuwan maupun civitas akademika karena *internet* memiliki potensi dalam menunjang kegiatan ilmiah sehingga tidak menutup kemungkinan dapat dijadikan sarana untuk meningkatkan produktivitas peneliti [RAH-03]. Banyaknya fungsi yang didapat dari *internet* ini disesuaikan dengan penggunaanya dimana terdapat mahasiswa, dosen dan karyawan yang dapat menggunakan fasilitas yang disediakan oleh Universitas Brawijaya ini.

Pola perilaku pengguna *internet* dapat dilihat melalui sebuah *file* yang mencatat segala informasi transaksi penggunaan *internet* yang disebut *access logs*. *Access logs* pada *proxy* Universitas Brawijaya berisi data-data akses pengguna *internet* yang melakukan transaksi dengan *webserver* di lingkungan Universitas Brawijaya. Data-data yang terdapat pada *access logs* tersebut dapat digali untuk mendapatkan informasi-informasi yang bermanfaat. Pengolahan data-data pada *access logs* pernah dilakukan dengan menggunakan metode *association rule mining*, penelitian ini membahas mengenai pola perilaku pengguna *internet* dengan studi kasus pada Fakultas MIPA Universitas Brawijaya. Data pada *access logs* dimanfaatkan untuk mendapatkan informasi-informasi pola perilaku pengguna *internet* berdasarkan waktu awal semester, tengah semester dan akhir semester [WIJ-13]. Pencarian pola perilaku pengguna *internet* dapat dilihat bukan

hanya berdasarkan awal, tengah atau akhir semester saja, melainkan juga dapat dilihat berdasarkan jam kerja dan jam istirahat atau jam-jam diluar jam kerja.

Mahasiswa Universitas Brawijaya yang berjumlah 52.376 orang pada tahun ajaran 2013/2014 merupakan populasi tertinggi dalam penggunaan *internet* di Universitas Brawijaya [PUB-13]. Jumlah populasi yang banyak menimbulkan permasalahan dalam alokasi pembagian *bandwith* yang tepat pada setiap jaringan *internet* agar seluruh mahasiswa dapat menggunakan fasilitas *internet* sesuai kebutuhannya. Bapak Alan selaku *head of system engineer* dari PPTI UB (Pengkajian dan Pengembangan Teknologi Informasi Universitas Brawijaya) menyatakan bahwa pada kenyataannya pembagian *bandwith* pada jaringan *internet* di Universitas Brawijaya hanya didasarkan pada perbedaan status mahasiswa, dosen atau karyawan tanpa melihat pola perilaku dari setiap fakultas yang ada. Pengambilan keputusan dalam pengalokasian *bandwith* bisa didapatkan dengan menganalisis pola perilaku pengguna *internet* tersebut. Setiap mahasiswa memiliki perilaku yang berbeda-beda dalam penggunaan *internet* tersebut tergantung dengan konsentrasi ilmu yang diambil dari tiap-tiap mahasiswa. Penggunaan *internet* antara fakultas eksak dan fakultas non eksak juga memiliki pola perilaku serta kebutuhan yang berbeda. Mahasiswa non eksak dan mahasiswa eksak menggunakan internet sebagai saluran informasi dengan kebutuhan pada mahasiswa non eksak (FISIP) 62% dan mahasiswa eksak (FARMASI) 80% [NOV-13]. Dari pernyataan tersebut dapat diambil kesimpulan bahwa mahasiswa konsentrasi ilmu eksak cenderung memiliki kebutuhan akses *internet* lebih besar dibandingkan dengan mahasiswa konsentrasi ilmu non eksak. Di Universitas Brawijaya terdapat 15 fakultas dengan 9 fakultas eksak dan sisanya merupakan fakultas non eksak.

Untuk mengolah data pada *access logs* dibutuhkan teknik yang dapat menghasilkan hubungan atau sebuah *rule* yang akurat mengingat jumlah data pada *access log* sangatlah besar. Data mining merupakan salah satu cara untuk membantu pengguna dengan menemukan hubungan dan menyajikannya dengan cara yang dapat dipahami sehingga hubungan tersebut dapat menjadi dasar pengambilan keputusan [MCL-07]. Dikarenakan data *access logs* merupakan data yang sekuensial atau berkelanjutan maka untuk mengolah data tersebut digunakan

sebuah metode yang bernama *sequential pattern mining* [BUD-09]. *Sequential pattern mining* digunakan untuk mencari data yang memiliki urutan, data tersebut bisa merupakan urutan transaksi [AGR-93].

Algoritma *sequential pattern discovery using equivalent classes* (SPADE) merupakan salah satu algoritma dari *sequential pattern mining* yang digunakan untuk menemukan pola agar mendapatkan informasi yang berguna. Algoritma SPADE menghasilkan performa yang jauh lebih baik dalam hal waktu komputasi dibandingkan algoritma-algoritma pencari *frequent sequences* sebelumnya seperti algoritma AprioriAll dan GSP (*Generalized Sequential Pattern*) [ARD-13]. Dengan latar belakang tersebut maka metode *sequential pattern mining* dengan algoritma SPADE diharapkan cocok untuk menemukan pola perilaku penggunaan *internet* di lingkungan Universitas Brawijaya. Sehingga judul pada penelitian ini adalah “**Sequential Pattern Mining Pada Pencarian Pola Perilaku Pengguna Internet Menggunakan Algoritma SPADE**”.

1.2 Rumusan Masalah

Dilihat dari uraian latar belakang maka dapat dirumuskan beberapa masalah sebagai berikut:

1. Bagaimana implementasi algoritma *sequential pattern discovery using equivalent classes* (SPADE) pada pencarian pola perilaku pengguna *internet* ?
2. Bagaimana tingkat kekuatan (*lift ratio*) rule yang dihasilkan algoritma *sequential pattern discovery using equivalent classes* (SPADE) dari data *access logs* ?

1.3 Batasan Masalah

Permasalahan pada penelitian dibatasi oleh:

1. Pola perilaku pengguna *internet* yang dimaksud dalam penelitian ini merupakan kecenderungan pengguna *internet* dalam mengakses suatu halaman *website*.
2. Pengguna *internet* pada penelitian ini merupakan civitas akademika dari fakultas-fakultas ilmu eksak di Universitas Brawijaya yaitu Fakultas Pertanian, Fakultas Peternakan, Fakultas Teknik, Fakultas Kedokteran,

Fakultas Perikanan dan Ilmu Kelautan, Fakultas Matematika dan Ilmu Pengetahuan Alam dan Fakultas Teknologi Pertanian.

3. Penelitian dilakukan dengan menggunakan sampel data *access logs* tahun 2011 pada *proxy* di Universitas Brawijaya Malang.
4. *Input* yang dibutuhkan oleh sistem ini antara lain nama fakultas, batas waktu transaksi, kategori waktu, besar minimum *support* dan besar minimum *confidence*.
5. Satu transaksi merupakan *log* dari aktivitas yang dilakukan oleh satu *user* dengan waktu transaksi yang berbeda-beda.

1.4 Tujuan

Tujuan dari penelitian ini adalah:

1. Untuk mengetahui implementasi algoritma *sequential pattern discovery using equivalent classes* (SPADE) pada pencarian pola perilaku pengguna *internet*.
2. Untuk mengetahui tingkat kekuatan (*lift ratio*) rule yang dihasilkan algoritma *sequential pattern discovery using equivalent classes* (SPADE) dari data *access logs*.

1.5 Manfaat

Manfaat dari penelitian ini adalah:

1. Bagi penulis:
 - 1.1 Sebagai media untuk mengaplikasikan ilmu yang telah didapat selama mengikuti perkuliahan di Teknik Informatika Universitas Brawijaya.
 - 1.2 Mendapatkan pemahaman mengenai penerapan algoritma *sequential pattern discovery using equivalent classes* (SPADE) pada lingkungan sekitar.
2. Bagi pengguna:
 - 2.1 Sebagai informasi atau modal dalam pengambilan keputusan seperti pengalokasian *bandwith* pada waktu-waktu tertentu.

1.6 Sistematika Penulisan

Penyusunan skripsi ini berdasarkan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat serta sistematika penulisan pada penelitian ini.

BAB II TINJAUAN PUSTAKA

Menguraikan teori-teori mengenai *internet*, *website*, *access logs*, data mining, *association rule*, *sequential pattern mining* dan *sequential pattern discovery using equivalent classes* (SPADE).

BAB III METODOLOGI DAN PERANCANGAN SISTEM

Membahas mengenai perancangan sistem, diagram alur proses dan contoh perhitungan manual.

BAB VI IMPLEMENTASI

Pada bab ini dijelaskan mengenai implementasi dari perancangan sistem yang telah dijabarkan pada bab sebelumnya.

BAB V PENGUJIAN DAN ANALISIS

Membahas mengenai hasil uji coba dari sistem yang telah diimplementasikan serta menganalisis hasil uji coba tersebut.

BAB VI PENUTUP

Pada bab ini memuat kesimpulan dari penelitian ini serta saran-saran untuk pengembangan penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian mengenai pencarian pola perilaku pengguna *internet* pernah dilakukan oleh Mila Febri Wijaya dengan studi kasus pada jurusan-jurusan di Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya. Pada penelitian tersebut digunakan metode *association rule* dengan algoritma Apriori untuk mencari pola perilaku pengguna *internet* pada waktu awal, tengah dan akhir semester. Penelitian tersebut dapat dikembangkan dengan mencari pola perilaku pengguna *internet* pada jam-jam tertentu, dimisalkan pada saat jam-jam kerja [WIJ-13].

Pada penelitian oleh Rikanda Ardiansyah, algoritma SPADE diterapkan pada data transaksi penjualan yang disimpulkan bahwa algoritma SPADE akurat untuk mencari pola sekuensial [ARD-13].

2.2 Pola Perilaku

Pola perilaku adalah cara masyarakat kelompok atau individu untuk bertindak atau berkelakuan yang sama. Pada kamus besar Bahasa Indonesia, kata pola memiliki arti betuk atau struktur sedangkan perilaku merupakan hasil pengalaman dan interaksi dengan lingkungan yang terwujud dalam pengetahuan sikap dan tindakan [MAU-07].

Pola perilaku merupakan salah satu yang membedakan antara individu atau kelompok dengan individu atau kelompok yang lain, hal tersebut dikarenakan pola perilaku merupakan sistem atau bentuk kecenderungan seseorang atau kelompok yang berinteraksi dengan serangkaian situasi [SOE-06].

2.3 Internet

Internet adalah singkatan dari *Interconnected Network*. *Internet* merupakan sebuah sistem komunikasi yang mampu menghubungkan jaringan-jaringan komputer di seluruh dunia. Berbagai jenis komputer dengan spesifikasi

yang berbeda-beda dapat saling berkomunikasi melalui *internet*. Beberapa bentuk jaringan yang berbeda-beda dapat saling bertukar informasi dan data melalui *internet* menggunakan seperangkat aturan yang disebut protokol TCP/IP [RAM-05].

Dalam perkembangannya *internet* dimanfaatkan sebagai media informasi dan komunikasi seperti tabloid atau koran *online* dan juga email. *Internet* sebagai media belajar seperti *e-learning* dan juga ensiklopedia *online*. *Internet* sebagai media hiburan seperti permainan *online*. *Internet* sebagai media bisnis dan perdagangan seperti *online shop* atau *e-commerce* [RAS-08].

2.4 Website

Website atau sering disebut dengan situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi dalam bentuk teks, gambar diam atau gerak, animasi, suara dan atau gabungan dari kesemuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Hubungan antara satu halaman *web* dengan *web* lainnya disebut dengan *hyperlink*, sedangkan teks yang dijadikan media penghubung disebut *hypertext* [HID-10].

Website adalah keseluruhan halaman-halaman *web* yang terdapat dalam sebuah *domain* yang mengandung informasi. *Domain* merupakan nama unik yang dimiliki oleh sebuah institusi sehingga dapat diakses melalui *internet*, seperti *lintau.com*, *yahoo.com*, *google.com* dan lain-lain [YUH-05].

2.5 URL (Universal Resource Locator)

URL adalah alamat sebuah *file* yang dapat diakses di *internet*. URL berisi nama protokol *internet* yang dibutuhkan untuk mengakses *file*, nama komputer yang berisi *file* tersebut, dan direktori di mana *file* tersebut berada [SUY-10].

Bagian pertama URL menunjukkan protokol misalnya “*http://*” atau “*https://*”. Bagian kedua URL menunjukkan alamat *server* dimana sumber daya tersebut terletak, misalnya “*www.microsoft.com*” untuk *website* Microsoft Corporation. Bagian ketiga URL adalah menunjukkan lokasi dan nama dokumen

atau program dalam *server* tersebut, misalnya “kb/deskapp/word/q1974.html” dimana “kb/deskapp/word” merupakan lokasi file sedangkan “q1974.html” adalah nama *file* [SUN-07].

2.6 DNS (*Domain Name Space*)

Domain Name Space merupakan sebuah hierarki pengelompokan domain berdasar nama yang terbagi menjadi beberapa bagian diantaranya [ART-03]:

1. *Root-Level Domains*

Domain ditentukan berdasarkan tingkatan kemampuan yang ada di struktur hirarki yang disebut dengan level. Level paling atas di hirarki disebut dengan *root domain*. *Root domain* di ekspresikan berdasarkan periode dimana lambang untuk root domain adalah (“.”).

2. *Top-Level Domains*

Contoh dari *top-level domains* diantaranya adalah com, edu, org, net, gov, mil, num, arpa, dan sebagainya. Top-level domains dapat berisi *second-level domains* dan *hosts*.

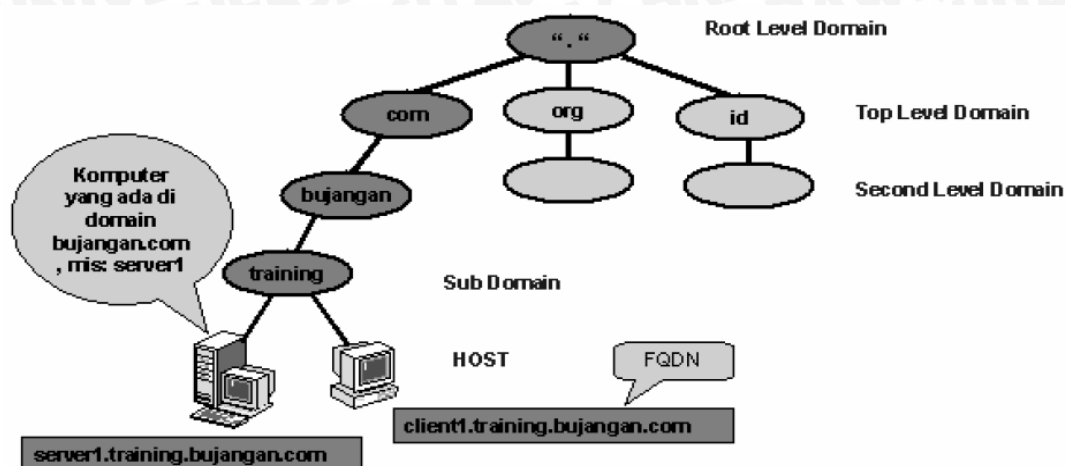
3. *Second-Level Domains*

Second-level domains dapat berisi *host* dan *domain* lain, yang disebut dengan *subdomain*. Untuk contoh: *Domain* Bujangan, bujangan.com terdapat komputer (*host*) seperti server1.bujangan.com dan *subdomain* training.bujangan.com. *Subdomain* training.bujangan.com juga terdapat komputer (*host*) seperti client1.training.bujangan.com.

4. *Host Names*

Domain name yang digunakan dengan *host name* akan menciptakan *fully qualified domain name* (FQDN) untuk setiap komputer. Sebagai contoh, jika terdapat fileserver1.detik.com, dimana fileserver1 adalah *host name* dan detik.com adalah *domain name*.

Gambaran perbedaan antara *root*, *top-level domain*, *second-level domain*, dan *host name* dijelaskan pada gambar 2.1.



Gambar 2.1 Hirarki DNS.

Sumber: [ART-03].

2.7 IP Address

IP address merupakan pengenalan yang digunakan untuk memberi alamat pada tiap-tiap komputer dalam jaringan [SYA-05]. Pada saat ini terdapat 2 versi *IP address*, yaitu IPv4 dan IPv6. Format *IP address* versi 4 adalah bilangan 32 bit yang tiap 8 bit-nya dipisahkan oleh tanda titik. Adapun format *IP address* dapat berupa bentuk biner (xxxxxxxx.xxxxxxxxxx.xxxxxxxxxx.xxxxxxxxxx dengan x merupakan bilangan biner 0 atau 1) atau dengan bentuk empat bilangan desimal yang masing-masingnya dipisahkan oleh titik [SYA-05]. Contoh *IP address* dengan bilangan desimal adalah 233.1.2.1.

Pada IPv4 terdapat dua bagian yaitu bagian adalah *network number* sedangkan bagian yang lain adalah *computer number (host number)*. Misal *IP address* 233.1.2.1, maka memiliki *network number* 233.1.2 dengan *host number* 1 [AMP-03]. Salah satu cara dalam pengalamatan *IP address* yaitu dengan menggunakan metode *classless addressing* yaitu dengan mengalokasikan *IP address* dalam notasi *Classless Inter Domain Routing (CIDR)* atau disebut juga dengan *network prefix* [SYA-05].

Dalam menuliskan *network prefix* suatu kelas *IP address* digunakan tanda miring (slash) “/” diikuti dengan angka yang menunjukkan panjang *network prefix* ini dalam bit. Contoh terdapat IP 167.205/18. Angka /18 merupakan notasi CIDR yang berarti *netmask* yang digunakan pada jaringan ini adalah 255.255.192.0

dengan jumlah maksimum *host* pada jaringan ini sebanyak 16.382 *node* [SYA-05].

2.8 Access Logs

Access logs adalah *file* yang mencatat semua akses transaksi yang dilakukan terhadap *web server*. Data yang disediakan oleh *web server logs* berisi informasi tentang *remotehost* (*ip address*), *rfc931*, *authuser*, *timestamp*, *request*, *status*, dan *size* serta jenis agen yang digunakan untuk mengakses *website* [PER-08]. Bentuk umum data *access log* sebagai berikut :

Tabel 2.1 Bentuk umum data *access logs*.

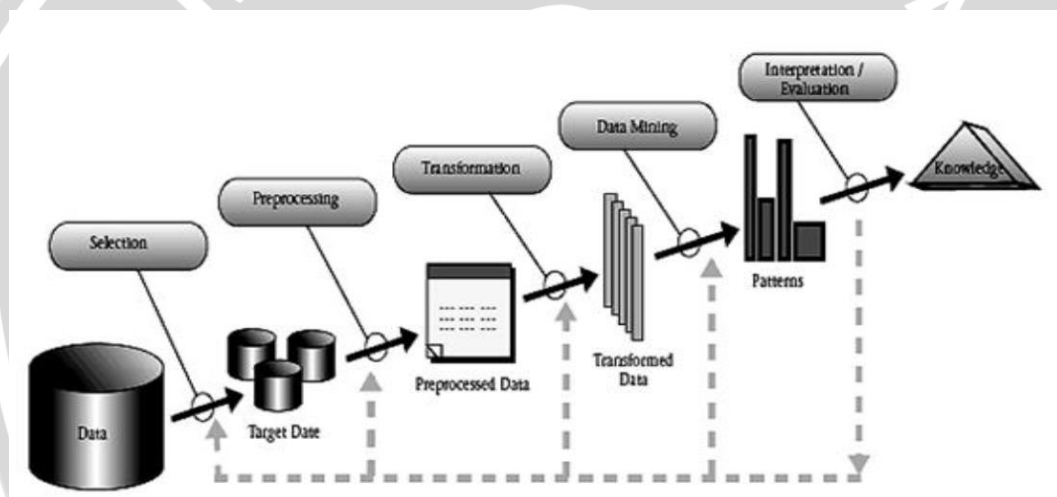
Nama Field	Deskripsi	Contoh
<i>Remotehost</i>	<i>Remote hostname</i> (atau <i>IP number</i> jika tidak terdapat <i>DNS hostname</i>)	127.0.0.1
<i>rfc931</i>	<i>remote login</i> untuk sebuah <i>user name</i> .	
<i>Authuser</i>	<i>username</i> yang melakukan otentifikasi terhadap dirinya	
<i>Date</i>	Waktu (tanggal dan jam) <i>request</i> berdasarkan zona waktu <i>web server</i>	[29/May/2007:14:50:04 +0700]
<i>Request</i>	Untuk data <i>request</i> , adalah data yang sebenarnya diperoleh dari aktivitas <i>user</i> . Data <i>request</i> ini dibagi menjadi tiga bagian, yaitu: <i>method</i> , alamat data yang diminta, dan jenis <i>protokol</i> yang digunakan.	GET / HTTP/1.1
<i>Status</i>	HTTP status kode yang dikembalikan ke <i>user</i> .	200
<i>Bytes</i>	Besarnya dokumen yang dikirim	12790
<i>Referrer</i>	URL yang dikunjungi <i>user</i> sebelum adanya permintaan terhadap URL yang baru	
<i>user_agent</i>	<i>Software</i> yang digunakan oleh <i>user</i>	Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)

Sumber : [PET-04].

2.9 Data Mining

Data mining merupakan solusi yang mampu menemukan kandungan informasi yang tersembunyi berupa pola dan aturan sekumpulan data yang besar agar mudah dipahami [HAN-01]. Data mining didefinisikan sebuah proses untuk menemukan hubungan, pola, dan tren baru yang bermakna dengan menyaring data yang sangat besar, yang tersimpan dalam penyimpanan, menggunakan teknik pengenalan pola seperti teknik statistik dan matematika [KUS-07].

Data mining dan *knowledge discovery in database* (KDD) merupakan istilah yang memiliki konsep yang berbeda akan tetapi saling berkaitan karena data mining adalah bagian dalam proses *knowledge discovery in database* (KDD) [KUS-07].



Gambar 2.2 Proses *knowledge discovery in database* (KDD).

Sumber [HAN-01].

Proses *knowledge discovery in database* (KDD) secara umum seperti gambar 2.2 adalah sebagai berikut [KUS-07]:

1. Data Selection

Merupakan tahap seleksi data, yang akan digunakan dalam proses data mining, dari sejumlah besar data operasional. Hasil dari seleksi data disimpan dalam suatu berkas terpisah dari *database* operasional.

2. *Preprocessing/Cleaning*

Pada tahap ini, dilakukan pembuangan duplikasi data, pemeriksaan data yang tidak konsisten, dan memperbaiki kesalahan pada data, serta memperkaya data yang sudah ada dengan data atau informasi eksternal.

3. *Transformation*

Tahap transformasi sangat bergantung pada jenis atau pola informasi yang akan dicari dalam basis data. Transformasi merupakan proses pengubahan sesuai format yang dibutuhkan untuk digunakan dalam proses data mining [HUD-10].

4. *Data Mining*

Merupakan tahap pencarian pola atau informasi dari data yang terpilih dengan menggunakan metode atau teknik tertentu. Ketepatan metode atau teknik yang dipilih sangat bergantung pada tujuan dari proses *knowledge discovery in database* (KDD) secara keseluruhan.

5. *Interpretation / Evaluation*

Pola informasi yang dihasilkan dari proses data mining perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya.

2.10 *Association Rule*

Association rule digunakan untuk menemukan hubungan diantara data atau bagaimana suatu kelompok data mempengaruhi suatu keberadaan data yang lain. Metode ini dapat membantu mengenali pola-pola tertentu di dalam kumpulan data yang besar. Dalam *association rule* suatu kelompok item dinamakan *itemset* [RUL-08]. *Association rule* merupakan pekerjaan untuk menentukan atribut yang akan didapatkan bersamaan. Tugas dari *association rule* adalah mencari aturan yang tersembunyi untuk mengukur hubungan antara dua atau lebih atribut [SOL-09].

Association rule merupakan bentuk dari data mining yang menyerupai proses menggali sesuatu yang sangat berharga dari basis data yang besar, dimana sesuatu yang sangat berharga ialah *rule* dengan nilai *confidence* yang lebih besar dari nilai batas minimum yang telah ditetapkan [KAN-03]. Dua tahap yang

dijalankan saat *association rule* adalah mencari kombinasi yang paling sering terjadi dari suatu itemset kemudian membangkitkan *association rule* dari frekuensi *itemset* yang telah dibuat sebelumnya [HAN-01].

2.11 Support

Support merupakan ukuran yang menunjukkan besar tingkat dominasi suatu *item* atau *itemset* keseluruhan transaksi [HAN-01]. *Support* digunakan untuk *rule* yang memiliki nilai kurang dari *threshold* atau minimum *support* yang telah ditentukan. Persamaan 2-1 merupakan persamaan untuk mendapatkan nilai *support*.

$$\text{Support } (A \rightarrow B) = \frac{\text{Jumlah transaksi itemset } (A \rightarrow B)}{\text{Total Transaksi}} \dots\dots \text{Persamaan (2-1)}$$

Sumber: [HAN-01].

2.12 Confidence

Confidence adalah ukuran yang menunjukkan hubungan antar 2 item secara kondisional [HAN-01]. *Confidence* adalah *ratio* antara jumlah transaksi yang meliputi semua item dalam *antecedent* dan *consequent* dengan jumlah transaksi meliputi semua *item* dalam *antecedent* [ROC-10]. Persamaan 2-2 merupakan persamaan untuk mendapatkan nilai *confidence*.

$$\text{Confidence } (A \rightarrow B) = \frac{\text{Jumlah transaksi itemset } (A \rightarrow B)}{\text{Jumlah antecendent A}} \dots\dots \text{Persamaan (2-2)}$$

Sumber: [HAN-01].

2.13 Sequential Pattern Mining

Sequential pattern mining pertama kali diperkenalkan oleh Agrawal dan Srikant pada tahun 1994. Pola sekuensial adalah pola yang menggambarkan urutan waktu dari peristiwa yang sering terjadi [HAN-01]. Diberikan satu set *sequence*, dimana setiap *sequence* terdiri dari satu set *item*, dan diberikan sebuah batasan minimum *support* yang ditentukan oleh pengguna, *sequential pattern*

mining adalah menemukan seluruh *sub-sequence* dimana frekuensi kemunculannya tidak lebih kecil dari minimum support [PEI-01].

Contoh pada tabel 2.2 merupakan data transaksi sekuensial yang memiliki 4 *sequence*. Perhatikan id 10 dengan *sequence* < a (abc) (ac) d (cf) > *Sequence* ini memiliki lima *event* yaitu a, abc, ac, d, dan cf yang terjadi sesuai urutannya. *Item* a dan c masing-masing muncul lebih dari satu kali pada *event* yang berbeda dalam *sequence id* 10 tersebut. *Item* a muncul tiga kali pada *sequence id* 10 dan memberikan kontribusi panjang sebesar 3 dari *sequence* tersebut. Jika terdapat sebuah *sub-sequence* $x = abc$, maka jika dilihat dari tabel 2.2, *sub-sequence* x memiliki nilai *support* sebesar 2 karena *sub-sequence* x terdapat pada *id* 10 dan *id* 30, sehingga *sub-sequence* x memenuhi minimum support.

Tabel 2.2 Contoh data transaksi.

Id	Sequence
10	< a (abc) (ac) d (cf) >
20	< (ad) c (bc) (ae) >
30	< (ef) (ab) (df) c b >
40	< e g (af) c b c >

Sumber : [HAN-01].

2.14 SPADE (*Sequential Pattern Discovery using Equivalent Classes*)

Sequential Pattern Discovery using Equivalent classes adalah salah satu algoritma *sequential pattern mining* yang menggunakan format data vertikal pada *database sequence*. Dalam format data vertikal, *database sequence* menjadi berbentuk kumpulan urutan yang formatnya [*itemset* : (*sequence_ID*, *event_ID*)], dengan kata lain, untuk setiap *itemset* akan disimpan *sequence identifier* dan *event identifier* yang berkoresponden. *Event Identifier* berguna sebagai *timestamp* atau penanda waktu dari *itemset* tersebut. Sepasang (*sequence_ID*, *event_ID*) untuk setiap *itemset* membentuk *ID_list* dari *itemset* tersebut [ARD-13].

Langkah-langkah algoritma SPADE dalam mencari *frequent sequence* kemudian menentukan *rule* dari *frequent sequence* tersebut adalah sebagai berikut [ARD-13]:

1. Menghitung *frequent 1-sequence*

Untuk mencari *frequent 1-sequence* yang harus dilakukan pertama kali adalah mendaftar semua *item* yang terdapat pada *database* transaksi. Setiap *item* akan dicari nilai *support*-nya dengan cara menjumlahkan SID yang memiliki *item* tersebut. Jumlah dari SID merupakan nilai *support* dari *item* tersebut. Setelah semua *item* ditemukan nilai *support*-nya maka, *item* yang memiliki nilai *support* lebih dari sama dengan nilai minimum *support* merupakan *item frequent 1-sequence*.

2. Menghitung *frequent 2-sequence*

Dalam mencari *frequent 2-sequence*, data yang digunakan adalah data dari *frequent 1-sequence*, sehingga tidak perlu mencari dari *sequence database* lagi. Untuk setiap masing-masing *frequent 1-sequence* akan digabungkan dengan semua *frequent 1-sequence* lainnya. Contohnya jika *1-sequence* A digabungkan dengan *1-sequence* B maka kemungkinan 2 *sequence* yang terjadi adalah A,B dimana A dan B muncul bersamaan dalam satu transaksi (satu SID, satu EID), A->B dimana item B muncul setelah item A, dan B->A dimana item A muncul setelah item B (satu SID, beda EID).

Untuk setiap masing-masing penggabungan *frequent 1-sequence* ini dilakukan pengecekan apakah eid dari *1-sequence* A sama dengan, kurang dari atau lebih dari eid *1-sequence* B. apabila sama maka id-listnya dimasukkan dalam *2-sequence* A,B. jika eid B lebih besar dari A maka id-listnya dimasukkan dalam *2-sequence* A->B dan jika eid A lebih besar dari B maka id-listnya dimasukkan dalam *2-sequence* B->A. kemudian seperti dalam *frequent 1-sequence id-list* akan ditambahkan supportnya untuk setiap masing-masing SID yang sebelumnya belum ditemui. Dari *2-sequence* itu kemudian dilakukan pengecekan apakah *support*-nya lebih dari nilai minimum *support*. Jika memenuhi syarat maka dimasukkan dalam *frequent 2-sequence*.

3. Menentukan *frequent k-sequence*

Setelah mencari *frequent 2-sequence*, untuk mencari *frequent sequence* berikutnya dilakukan proses yang sama, yaitu mencari *frequent k-sequence*. Untuk mencari *frequent k-sequence* ini dilakukan join pada *frequent (k-1) sequence* yang memiliki *prefix* yang sama. Contohnya untuk mencari 3-*sequence* maka perlu digabungkan *frequent sequence* dari 2-*sequence* yang memiliki *prefix* yang sama, untuk mencari 4-*sequence* maka, perlu digabungkan *frequent sequence* dari 3-*sequence* yang memiliki *prefix* yang sama, dan seterusnya. Untuk mencari *prefix frequent (k-1), sequence* akan menghilangkan item terakhir dari *sequence* tersebut. Contoh jika terdapat 4-*sequence* $A \rightarrow B \rightarrow C \rightarrow D$, maka *prefix*-nya adalah $A \rightarrow B \rightarrow C$. untuk setiap penggabungan ini ada 3 kemungkinan hasil:

- Jika A,B digabungkan dengan A,C maka kemungkinan hasilnya hanya A,B,C.
- Jika A,B digabungkan dengan $A \rightarrow C$, maka kemungkinan hasilnya hanya $A, B \rightarrow C$
- Jika $A \rightarrow B$ digabungkan dengan $A \rightarrow C$, maka ada 3 kemungkinan hasil: $A \rightarrow B, C$, dan $A \rightarrow B \rightarrow C$ dan $A \rightarrow C \rightarrow B$

Dari setiap kemungkinan ini kemudian mengecek nilai *support*-nya apakah memenuhi minimal *support* atau tidak. Jika ya maka *sequence* itu termasuk dalam *frequent k-sequence*. Pencarian *frequent sequence* dihentikan apabila tidak ada *frequent(k-1) sequence* yang bisa digabungkan atau sudah tidak ditemukan *frequent k-sequence* lagi.

4. Pembentukan Rule

Setelah ditemukan semua *frequent sequence*, ditentukan *rule* dari *sequence-sequence* tersebut. 1-*sequence* tidak digunakan untuk membentuk *rule* karena hanya terdiri dari 1 item. Untuk 2-*sequence* yang menjadi *antecedent* adalah item pertama dan *consequent*-nya adalah item kedua. Contohnya untuk *sequence* $A \rightarrow B$ maka *rule* yang dibentuk adalah $A \Rightarrow B$. sedangkan untuk *sequence* yang panjangnya lebih dari 2 atau *k-sequence*, yang dijadikan *consequent* adalah item terakhir, sedangkan *antecedent*nya adalah semua item

sebelum item terakhir. Contohnya pada 4-sequence A->B->C->D, maka *rule* yang dihasilkan adalah A->B->C=>D. Untuk masing-masing *rule* dihitung nilai *confidence*-nya serta nilai *support*. *Rule* yang memenuhi atau melebihi dari nilai minimum *confidence* dan minimum *support* merupakan *rule* yang digunakan dan kemudian untuk dihitung lift rasionya untuk menentukan *rule* mana yang paling kuat.

2.15 Lift Ratio

Lift ratio dapat digunakan untuk mengetahui kuat tidaknya sebuah *rule*. *Lift ratio* dapat dihitung dengan cara membandingkan nilai *confidence* dengan nilai *benchmark confidence*. Nilai dari *benchmark confidence* setiap *rule* didapatkan dari nilai jumlah transaksi *consequent* dibagi dengan jumlah transaksi [LEU-07]. Rumus dari *benchmark confidence* dapat dilihat pada persamaan (2-3).

$$\text{Benchmark Confidence} = \frac{nc}{n} \dots \dots \dots \text{Persamaan (2-3)}$$

nc = Jumlah transaksi item yang menjadi *consequent*.

n = Jumlah transaksi basis data

Sumber: [SAN-07].

$$\text{Lift Ratio} = \frac{\text{Confidence}}{\text{Benchmark Confidence}} \dots \dots \dots \text{Persamaan (2-4)}$$

Sumber: [SAN-07].

Pada persamaan (2-4) dijelaskan cara untuk mendapatkan nilai *lift ratio*. Apabila nilai *lift ratio* lebih besar dari 1, maka menunjukkan adanya manfaat dari aturan tersebut. Lebih tinggi nilai *lift ratio* maka lebih besar kekuatan asosiasinya [SAN-07].

BAB III

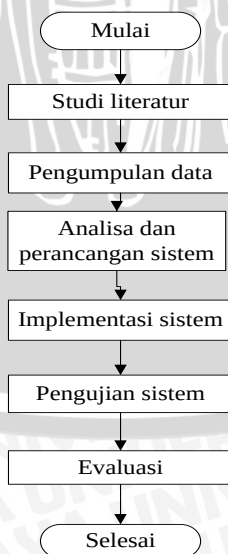
METODOLOGI DAN PERANCANGAN SISTEM

3.1 Metodologi

Pada bab ini akan dijabarkan mengenai metodologi serta perancangan dari sistem dalam pencarian pola perilaku pengguna *internet* menggunakan algoritma SPADE. Langkah-langkah yang digunakan untuk menggali data *access logs* dengan menggunakan algoritma SPADE meliputi:

1. Melakukan studi literatur mengenai *sequential pattern mining* dan algoritma SPADE.
2. Mengumpulkan data-data yang diperlukan, dalam hal ini yaitu data *access logs*.
3. Melakukan perancangan sistem untuk menggali data *access logs* dengan menggunakan algoritma SPADE.
4. Melakukan implementasi sistem berdasarkan perancangan yang dilakukan sebelumnya.
5. Pengujian terhadap sistem dilakukan dengan menggunakan data *access logs*.
6. Mengevaluasi hasil pengujian sistem.

Langkah-langkah pada penelitian ini dapat digambarkan sebagai bagan *flowchart* seperti pada Gambar 3.1.



Gambar 3.1. *Flowchart* langkah-langkah kerja penelitian.

3.2 Pengumpulan Data

Pengumpulan data dilakukan dengan menggunakan pengumpulan data lapangan. Data tersebut merupakan data *access logs* pada *proxy* di Universitas Brawijaya. Data yang digunakan merupakan sampel data *access logs* pada tahun 2011. Tabel 3.1 menunjukkan beberapa contoh data *access logs* pada *proxy* di Universitas Brawijaya.

Tabel 3.1. Contoh data *access logs* pada *proxy* di Universitas Brawijaya.

NO	DATA
1	1304229106.360 78 172.17.170.175 TCP_MISS/200 2057 GET http://block.opendns.com/? - DIRECT/67.215.80.135 text/html
2	1304187493.916 224 172.17.110.55 TCP_MISS/304 247 GET http://profile.ak.fbcdn.net/hprofile-ak-snc4/195267_100001042550313_6874764_q.jpg - DIRECT/173.223.52.193 image/jpeg
3	1304187493.935 4 172.17.51.116 TCP_HIT/200 5041 GET http://clkads.com/adServe/script/effects_1.0.5.js - NONE/- application/x-javascript
4	1304187493.969 448 172.17.102.147 TCP_MISS/200 404 GET http://cdn-test.wefi.com/test.html - DIRECT/208.93.141.180 text/html
5	1304187493.978 220 172.17.110.55 TCP_MISS/304 247 GET http://profile.ak.fbcdn.net/hprofile-ak-snc4/41716_100001033441672_5107070_q.jpg - DIRECT/173.223.52.178 image/jpeg

Sumber: PPTI-UB.

3.3 Analisis dan Perancangan Sistem

3.3.1 Deskripsi Sistem

Sistem yang akan dibuat merupakan sebuah aplikasi desktop yang dibangun menggunakan bahasa pemrograman Java. Sistem ini dikembangkan untuk melakukan analisa terhadap data *access logs* pada fakultas-fakultas ilmu eksak di Universitas Brawijaya. Analisa terhadap data *access logs* tersebut digunakan untuk mengetahui pola perilaku pengguna *internet* dalam mengakses

situs-situs pada saat terkoneksi dengan jaringan *internet* di lingkungan Universitas Brawijaya.

Parameter yang digunakan dalam analisa adalah nilai minimum *support*, nilai minimum *confidence*, nama fakultas, batas waktu transaksi serta kategori jam. Nama fakultas yang dijadikan sebagai parameter dapat dilihat pada Tabel 3.2.

Tabel 3.2. Nama-nama fakultas ilmu eksak di Universitas Brawijaya.

NO	NAMA FAKULTAS
1	Fakultas Pertanian
2	Fakultas Peternakan
3	Fakultas Teknik
4	Fakultas Kedokteran
5	Fakultas Perikanan dan Kelautan
6	Fakultas Matematika dan Ilmu Pengetahuan Alam
7	Fakultas Teknologi Pertanian

Sumber: Perancangan.

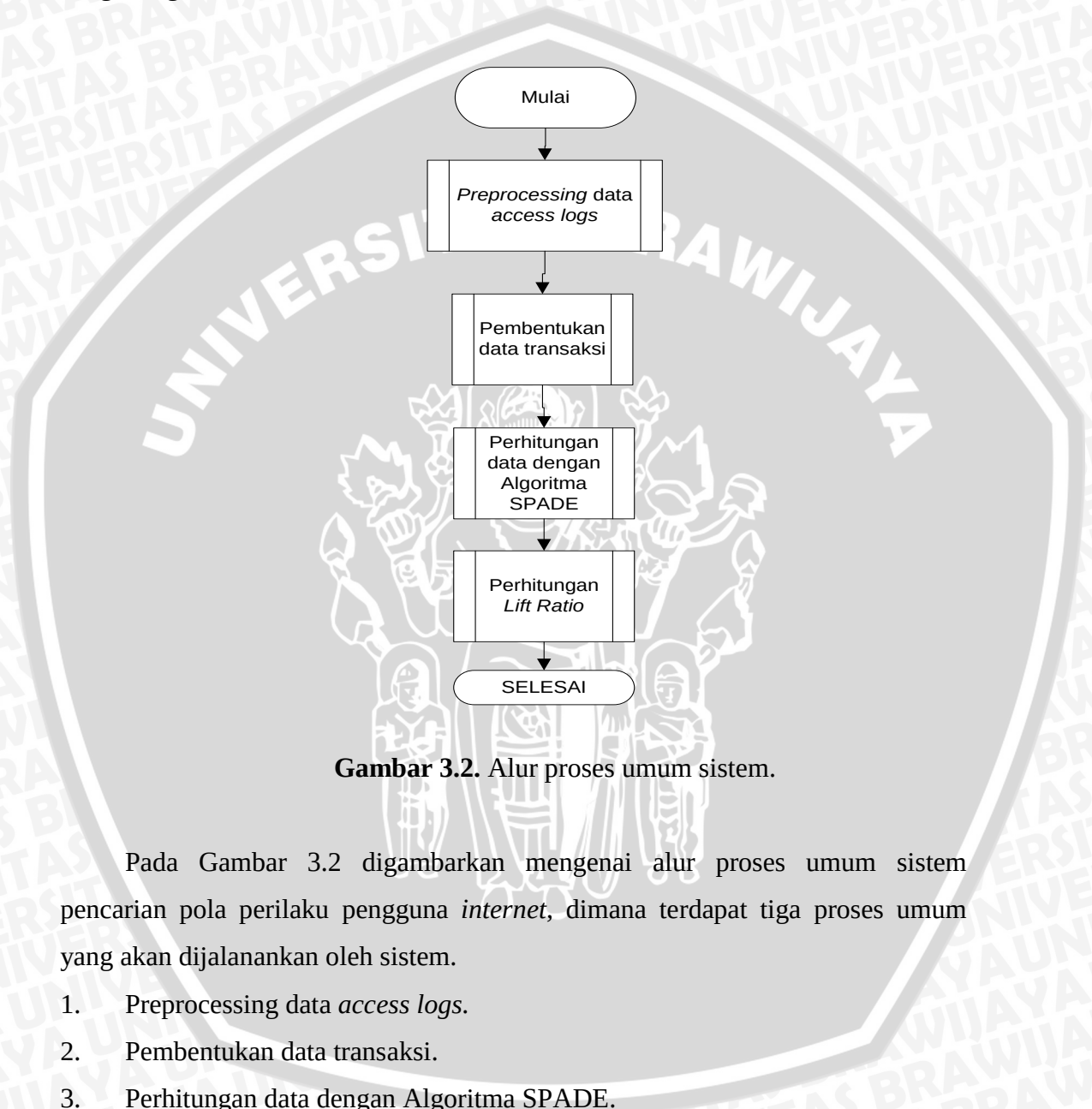
Batas waktu transaksi yang akan digunakan sebagai parameter merupakan lama waktu pengguna *internet* mengakses *internet*. Batas waktu yang digunakan memiliki satuan menit. Sedangkan kategori jam yang dimaksud sebagai parameter merupakan waktu disaat pengguna *internet* mengakses *internet* yang digolongkan menjadi tiga yaitu jam kerja, jam istirahat, dan diluar jam kerja dan istirahat. Pembagian kategori jam dijabarkan pada Tabel 3.3 dibawah ini.

Tabel 3.3. Pembagian kategori waktu.

NO	KATEGORI	RENTANG WAKTU
1	Jam Kerja	08.00 – 12.00 dan 13.00 – 16.00
2	Jam Istirahat	12.00 - 13.00
3	Diluar jam kerja	00.00 – 08.00 dan 16.00 – 00.00

Sumber: Perancangan.

Hasil yang dikeluarkan oleh sistem ini merupakan *rule-rule* dari perhitungan pada data *access logs* yang dipengaruhi oleh parameter-parameter yang telah disebutkan diatas menggunakan metode *sequential pattern mining* dengan algoritma SPADE.



Gambar 3.2. Alur proses umum sistem.

Pada Gambar 3.2 digambarkan mengenai alur proses umum sistem pencarian pola perilaku pengguna *internet*, dimana terdapat tiga proses umum yang akan dijalankan oleh sistem.

1. Preprocessing data *access logs*.
2. Pembentukan data transaksi.
3. Perhitungan data dengan Algoritma SPADE.
4. Perhitungan *lift ratio*.

3.3.2 Batasan Sistem

Batasan dari sistem yang dibangun ini adalah:

1. Input yang dibutuhkan adalah nilai minimum *support*, nilai minimum *confidence*, nama fakultas, waktu transaksi, dan bulan.
2. Sistem yang dibangun merupakan aplikasi *desktop* dengan bahasa pemrograman Java.

3.4 Perancangan Proses

Perancangan proses digunakan sebagai pedoman dalam pembangunan sistem pencarian pola perilaku pengguna *internet* pada data *access logs* menggunakan algoritma SPADE. *Flowchart* serta alur proses dibutuhkan untuk memudahkan dalam membangun sistem tersebut.

3.4.1 Preprocessing Data Access Logs

Proses pertama yang dilakukan oleh sistem ini adalah *preprocessing* data *access logs* dengan menyaring informasi-informasi yang dibutuhkan dan membuang data-data yang tidak dibutuhkan. Informasi-informasi yang dibutuhkan adalah *IP address*, *URL destination*, waktu akses dan *MIME Type*.

Pada *IP address* akan dilakukan *filtering* untuk mendapatkan *range IP address* pada fakultas-fakultas ilmu eksak. Tabel 3.4 merupakan alokasi *IP address* pada fakultas-fakultas ilmu eksak di Universitas Brawijaya.

Tabel 3.4. Alokasi *IP address* Fakultas-fakultas Ilmu Eksak di Universitas Brawijaya.

NO	FAKULTAS	RANGE IP ADDRESS
1	Fakultas Pertanian	172.17.40.0 – 172.17.49.0
2	Fakultas Peternakan	172.17.50.0-172.17.59.0
3	Fakultas Teknik	172.17.60.0 – 172.17.79.0
4	Fakultas Kedokteran	172.17.160.0 – 172.17.169.0
5	Fakultas Perikanan dan Kelautan	172.17.80.0 – 172.17.89.0
6	Fakultas Matematika dan Ilmu Pengatuhan Alam	172.17.100.0 – 172.17.109.0
7	Fakultas Teknologi Pertanian	172.17.110.0 – 172.17.119.0

Sumber: PPTI-UB.

Pada *URL destination* dilakukan *filtering* untuk mengambil nama *host* tanpa mempedulikan lokasi file yang diakses. Sedangkan *MIME Type* digunakan untuk mengetahui jenis *request* yang diakses dengan cara menyaring data *file multimedia* (gambar, *icon*, animasi, suara dan video), *client-side scriptfile*, dan *cascading style sheet file*. Informasi tersebut diabaikan karena merupakan bagian dari suatu *request* terhadap sebuah halaman *web* [SIG-06].

Pada tabel 3.1 merupakan contoh data *access logs* pada *proxy* Universitas Brawijaya yang akan dilakukan *filtering*. Data tersebut memiliki elemen-elemen yang akan digunakan sebagai informasi untuk membentuk data transaksi dan melakukan perhitungan. Elemen-elemen pada data *access logs* tersebut dijelaskan pada Tabel 3.5

Tabel 3.5. Elemen-elemen pada Data Access Logs.

NAMA ELEMEN	ISI ELEMEN
DateTime	1304187493.978
Port	220
Host	172.17.110.55
Code/Status	TCP_MISS/304
Byte	247
Method	GET
URL	http://profile.ak.fbcdn.net/hprofile-ak-snc4/41716_100001033441672_5107070_q.jpg
Authuser	-
PeerStatus/PeerHost	DIRECT/173.223.52.178
MIME Type	image/jpeg

Sumber: PPTI-UB.

Elemen yang digunakan sebagai informasi untuk membentuk data transaksi dan juga untuk digunakan dalam perhitungan adalah *DateTime*, *Host*, *URL* dan *MIMEType*. Setelah melakukan proses *filtering* data pada tabel 3.1 maka, hanya dua data saja yang digunakan untuk membentuk data transaksi. Pada Tabel 3.6 merupakan data *access logs* setelah dilakukan *filtering*. Data hasil *filtering* menyisakan data dengan nomer 1 dan 4, sedangkan data lainnya tidak digunakan atau tidak disimpan karena status dari *MIME type* bukan merupakan status *request* dari halaman *web*.

Tabel 3.6. Hasil *Filtering* data *Access Logs*.

NO	DATA
1	1304229106.360 172.17.170.175 block.opendns.com text/html
4	1304187493.969 172.17.102.147 cdn-test.wefi.com text/html

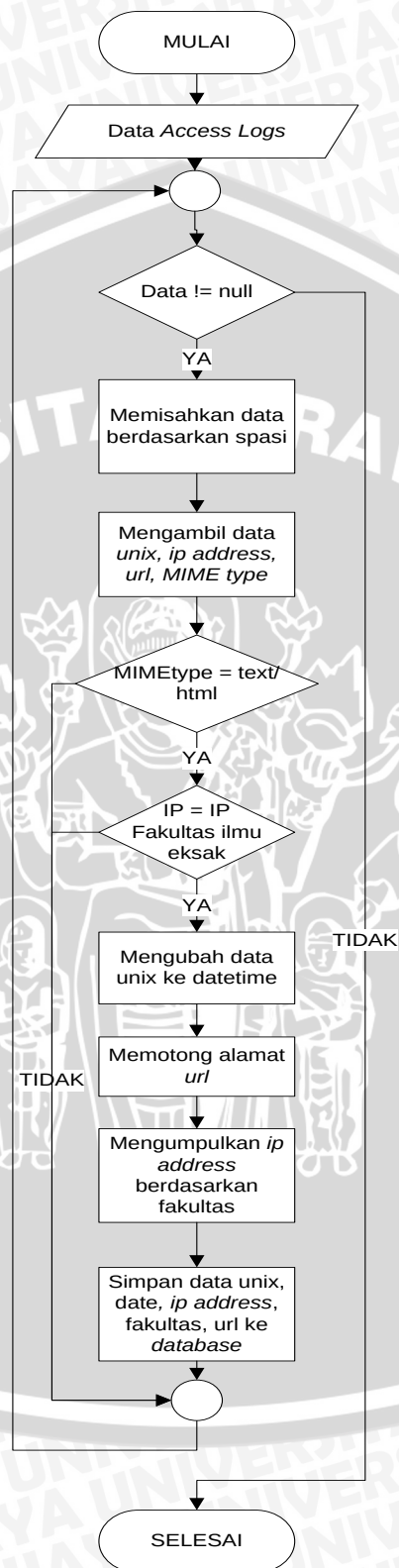
Sumber: Perancangan.

Langkah-langkah pada proses *preprocessing* data adalah sebagai berikut:

1. Memasukkan data berupa data *access logs*. Bentuk data *access logs* berupa *file* dengan ekstensi *.txt*.
2. Melakukan pembacaan data *access logs*. Proses pembacaan dilakukan dengan membaca setiap baris hingga tidak ada baris lagi yang dibaca. Setiap baris pada data *access logs* merupakan catatan satu transaksi.
3. Memisahkan data *access logs* berdasarkan spasi pada setiap barisnya.
4. Mengambil data *unix*, *IP address*, *url* dan *MIME type*.
5. Memeriksa nilai *MIME type* harus berupa halaman *web*. Hal tersebut dilihat dari status *MIME type* yang harus bertuliskan “text/html”. Jika *MIME type* tidak berupa halaman *web* maka, proses dikembalikan pada proses 2 dengan membaca baris selanjutnya atau transaksi selanjutnya.
6. Memeriksa *IP address*, apakah *IP* tersebut termasuk dalam range *IP address* fakultas-fakultas ilmu eksak di Universitas Brawijaya atau tidak. Jika tidak maka, proses dikembalikan pada proses 2 dengan membaca baris selanjutnya atau transaksi selanjutnya.
7. Merubah data *unix* menjadi *datetime*.
8. Melakukan *filtering* pada alamat *url* dengan menghilangkan *domain*, serta lokasi *file*.
9. Menyimpan *datetime*, *IP address*, *url* dan *MIME type* pada *database*.
10. Proses kembali ke proses 2 hingga tidak ada lagi baris yang dibaca atau tidak ada lagi data transaksi yang dibaca.

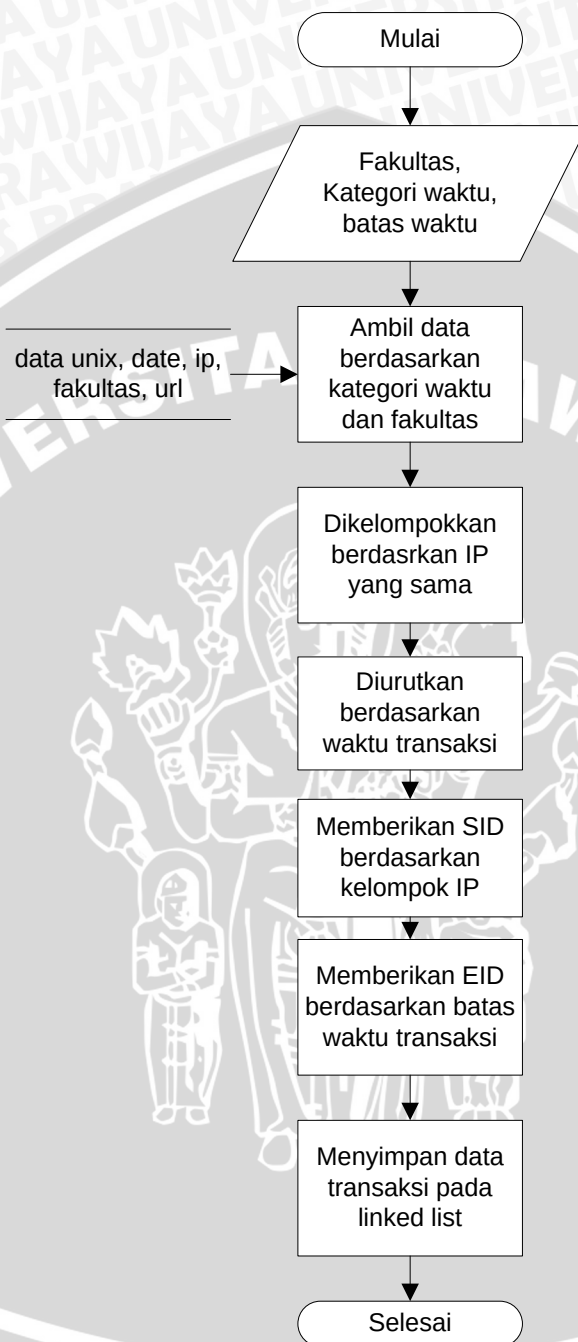
Flowchart proses dalam *preprocessing* data dapat dilihat pada Gambar

3.3.



Gambar 3.3. Flowchart proses preprocessing data.

3.4.2 Pembentukan Data Transaksi



Gambar 3.4. Flowchart proses pembentukan data transaksi.

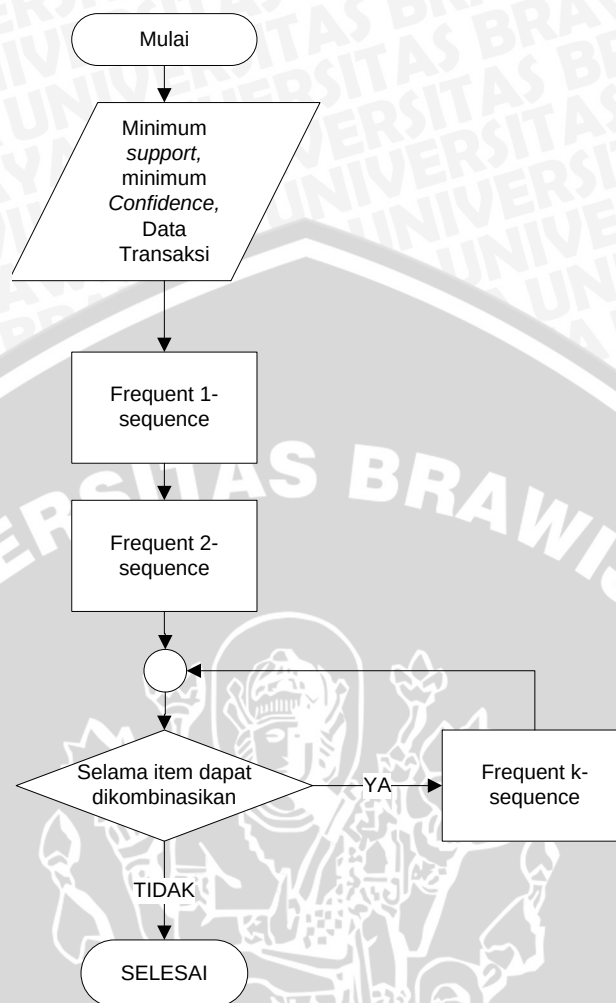
Pembentukan data transaksi digunakan untuk mengelompokkan *IP address* menjadi *SID (sequence identifier)* serta mengelompokkan transaksi berdasarkan batas waktu transaksi menjadi *EID (event identifier)*. Pada awal pembentukan data transaksi, diperlukan masukan berupa kategori waktu, nama fakultas dan batas waktu yang dibutuhkan dalam satu transaksi. Selanjutnya akan dibentuk kumpulan transaksi yang dikelompokkan berdasarkan rentang waktu yang telah ditentukan dan juga pelaku transaksi dalam hal ini adalah *user* yang diwakili oleh *IP address*. Data transaksi yang telah terbentuk akan digunakan sebagai pedoman dalam perhitungan SPADE pada proses berikutnya. *Flowchart* proses dalam pembentukan data transaksi terdapat pada Gambar 3.4.

Langkah-langkah dalam pembentukan data transaksi adalah sebagai berikut:

1. Menentukan nama fakultas, kategori waktu, dan batas waktu yang digunakan untuk membentuk data transaksi.
2. Mengambil data pada *database* berdasarkan fakultas dan kategori waktu yang telah ditentukan.
3. Mengelompokkan data berdasarkan *IP address* yang sama.
4. Memberikan pengkodean *SID (sequence identifier)* berupa nomor pada *IP address*.
5. Mengelompokkan data berdasarkan batas waktu serta diurutkan untuk diberikan pengkodean *EID (event identifier)* berupa nomor sesuai urutan waktu akses transaksi.
6. Menyimpan data transaksi dengan memanfaatkan teknik *penyimpanan linked list*.

3.4.3 Perhitungan SPADE

Perhitungan SPADE dilakukan ketika data transaksi telah terbentuk. Untuk lebih jelasnya mengenai tahap-tahap perhitungan SPADE dapat dilihat pada Gambar 3.5.

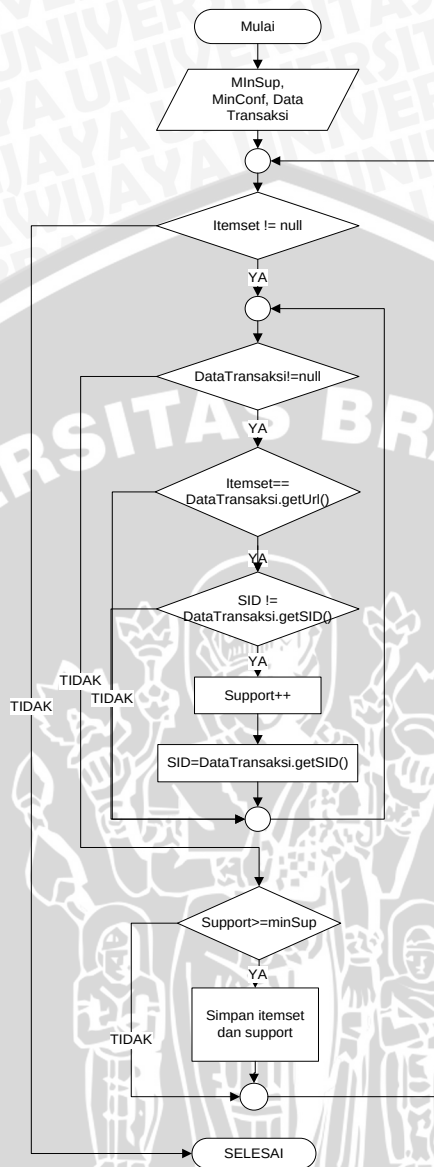


Gambar 3.5. Flowchart perhitungan SPADE.

Perhitungan SPADE dilakukan dengan tahap-tahap sebagai berikut:

1. Memasukkan nilai minimum *support*, nilai minimum *confidence* dan juga data transaksi.
2. Menemukan *itemset frequent 1-sequence*.
3. Menemukan *itemset frequent 2-sequence* dengan cara mengkombinasikan sesama *itemset frequent 1-sequence* yang telah terbentuk pada proses sebelumnya.
4. Jika *itemset* masih dapat dikombinasikan dan terdapat pada data transaksi maka dilakukan pembentukan *frequent k-sequence*. Jika tidak maka berlanjut proses berakhir.

3.4.3.1 Pembentukan 1-Sequence



Gambar 3.6. Flowchart pembentukan frequent 1-sequence.

Gambar 3.6 merupakan langkah-langkah pembentukan frequent 1-sequence yang merupakan langkah awal dalam perhitungan dengan menggunakan algoritma SPADE. Langkah-langkah pembentukan 1-sequence adalah sebagai berikut:

1. Memasukkan nilai minimum support, minimum confidence serta data transaksi
2. Melakukan pembacaan data itemset yang terdiri dari item-item yang terdapat pada data transaksi.

3. Melakukan pembacaan pada data transaksi, selama data transaksi masih ada.
4. Menyamakan isi *item* pada *itemset* dan juga *item* pada data transaksi. Jika sama maka proses berlanjut jika tidak maka proses kembali ke proses nomer 3.
5. Menyamakan apakah pada SID yang sama terdapat *item* yang sama. Jika tidak maka proses dilanjutkan jika iya maka proses kembali ke proses nomer 3.
6. Menambah nilai *support* untuk *item* yang sedang ditunjuk.
7. Melakukan pengecekan nilai *support* lebih dari sama dengan nilai minimum *support*. Jika tidak kembali ke proses nomer 2. Jika iya maka *item* dan nilai *support* disimpan pada *linked list*.

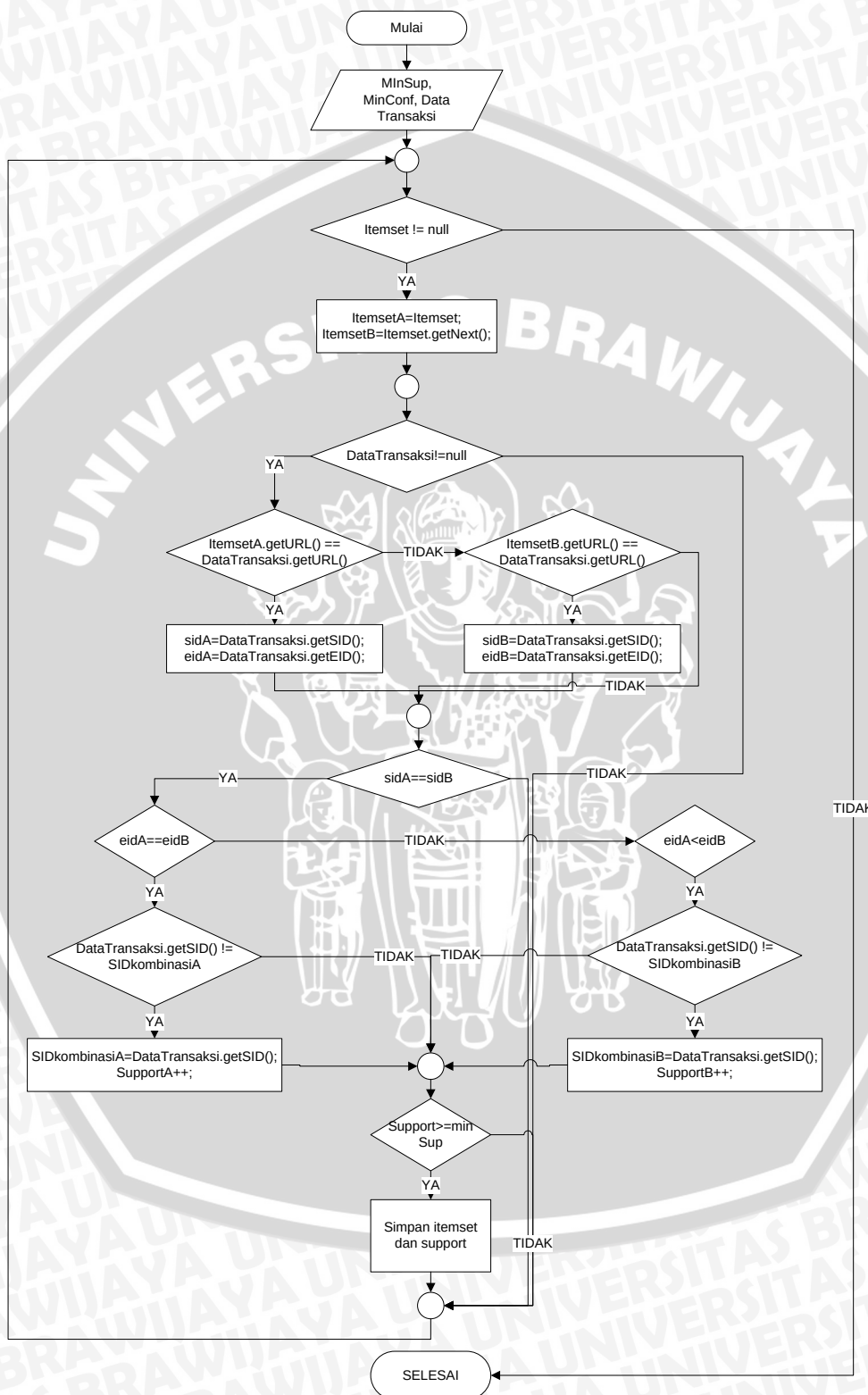
3.4.3.2 Pembentukan 2-Sequence

Setelah menemukan *frequent 1-sequence* maka, dilanjutkan dengan pembentukan *frequent 2-sequence*. Berikut merupakan langkah-langkah pembentukan *frequent 2-sequence*:

1. Membaca data *itemset frequent 1-sequence* selama data tersebut masih ada.
2. Membaca data transaksi selama data tersebut masih ada, jika sudah tidak ada atau habis maka proses kembali ke proses nomer 1.
3. Melakukan pengecekan apakah *item A* (*item* pertama) sama dengan *item* pada data transaksi, jika tidak maka melakukan pengecekan apakah *item B* (*item* kedua) sama dengan *item* pada data transaksi, jika tidak maka proses kembali ke proses nomer 2. Jika iya maka melakukan pengambilan data SID dan EID.
4. Melakukan pengecekan apakah SID *item A* sama dengan SID *item B*. jika iya maka, proses berlanjut. Jika tidak maka, proses kembali ke proses nomer 2.
5. Melakukan pengecekan apakah EID *item A* sama dengan EID *item B*, jika iya maka jenis kombinasi A,B. jika tidak maka dilakukan pengecekan apakah eid *item A* lebih kecil dari eid *item B*. jika iya maka kombinasi menjadi A -> B.
6. Melakukan penambahan nilai *support* pada setiap kombinasi yang sama ketika menemukan jenis kombinasi yang sesuai pada SID yang berbeda.
7. Melakukan pengecekan terhadap nilai *support*. Jika nilai *support* lebih besar sama dengan nilai minimum *support* maka kombinasi *item* dan nilai *support* disimpan pada *frequent 2-sequence*.

Langkah-langkah pembentukan *frequent 2-sequence* dapat dilihat pada Gambar

3.7.



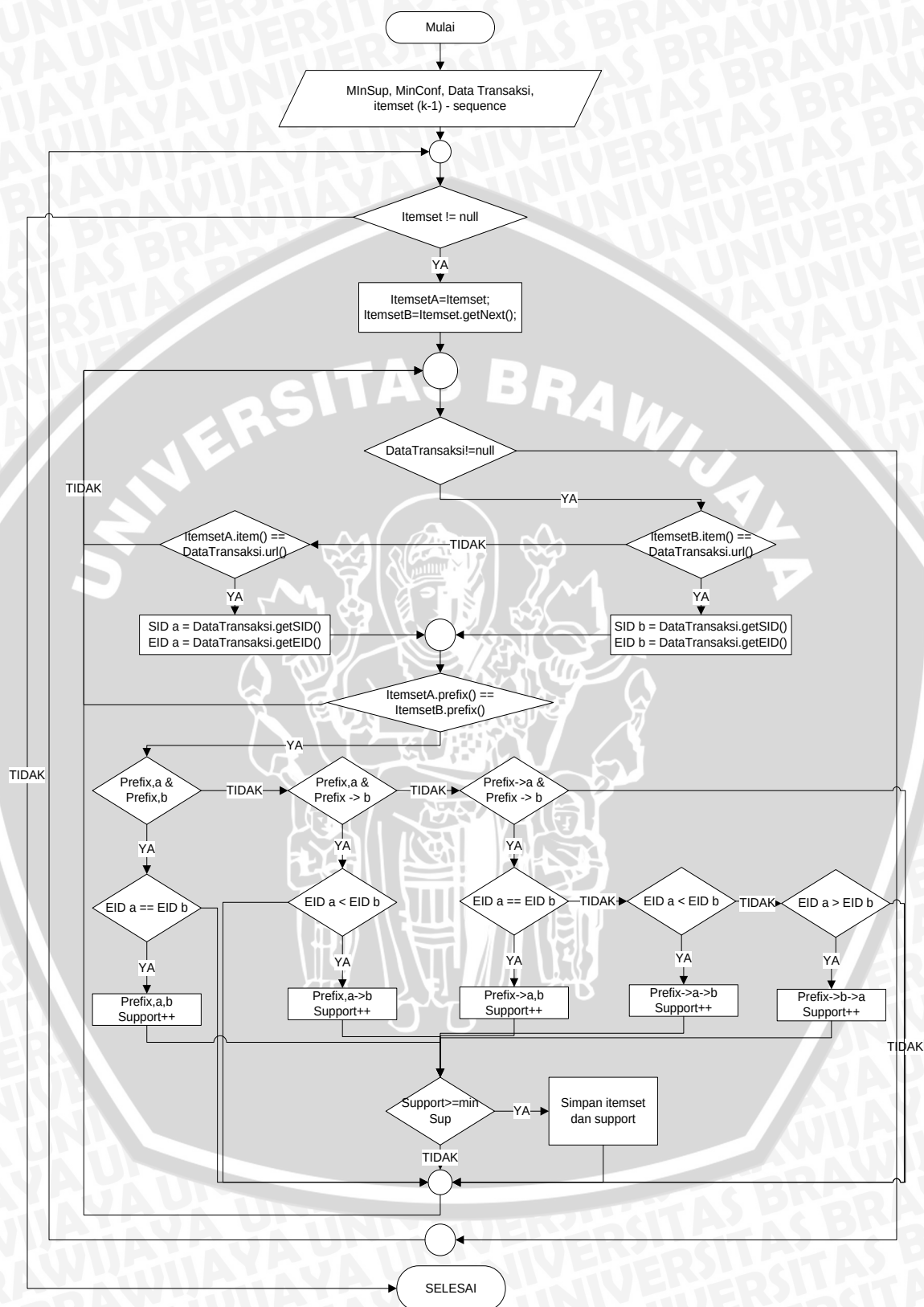
Gambar 3.7. Flowchart pembentukan *frequent 2-sequence*.

3.4.3.3 Pembentukan k-Sequence

Setelah menemukan *frequent 2-sequence* maka, dilanjutkan dengan pembentukan *frequent k-sequence*. Dalam pembentukan *frequent k-sequence* dibutuhkan *frequent (k-1)-sequence* untuk dikombinasikan. Berikut merupakan langkah-langkah pembentukan *frequent k-sequence*:

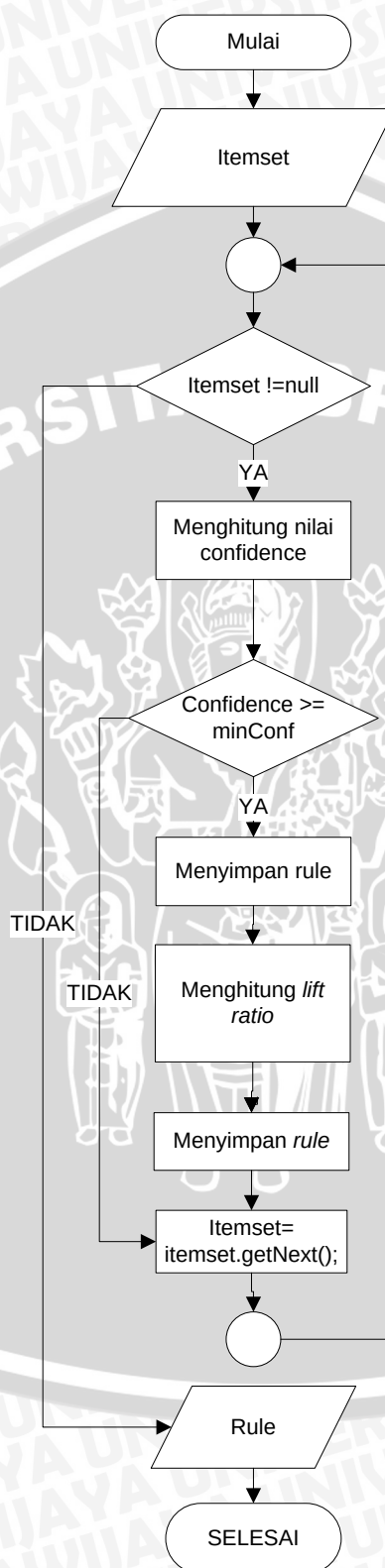
1. Membaca data itemset *frequent (k-1)-sequence* selama data tersebut masih ada.
2. Membaca data transaksi selama data tersebut masih ada, jika sudah tidak ada atau habis maka proses kembali ke proses nomer 1.
3. Melakukan pengecekan apakah *item* sama dengan *item* pada data transaksi.
4. Melakukan pengecekan apakah *prefix item* A sama dengan *prefix item* B. Jika iya maka, proses berlanjut. Jika tidak maka, proses kembali ke proses nomer 2.
5. Melakukan pengecekan hubungan antara *prefix*, a dan b:
 - a. Jika *prefix*,a dan *prefix*,b maka melakukan pengecekan apakah EID a = EID b. Jika iya maka hasil kombinasi adalah *prefix*,a,b.
 - b. Jika *prefix*,a dan *prefix* -> b maka melakukan pengecekan apakah EID a < EID b. Jika iya maka hasil kombinasi adalah *prefix*,a -> b.
 - c. Jika *prefix*->a dan *prefix*->b maka hasil kombinasinya ada tiga yaitu:
 - i. Jika EID a = EID b maka, *prefix* ->a,b.
 - ii. Jika EID a < EID b maka, *prefix* ->a->b.
 - iii. Jika EID a > EID b maka, *prefix* ->b->a
6. Melakukan penambahan nilai *support* pada setiap kombinasi yang sama ketika menemukan jenis kombinasi yang sesuai pada SID yang berbeda.
7. Melakukan pengecekan terhadap nilai *support*. Jika nilai *support* lebih besar sama dengan nilai minimum *support* maka kombinasi item dan nilai *support* disimpan pada *frequent k-sequence*.

Langkah-langkah pembentukan *frequent k-sequence* dapat dilihat pada Gambar 3.8.



Gambar 3.8. Flowchart pembentukan frequent k-sequence.

3.4.4 Perhitungan lift ratio



Gambar 3.9. Flowchart perhitungan lift ratio.

Untuk menemukan *rule* yang kuat dari hasil perhitungan SPADE, diperlukan adanya perhitungan *lift ratio*. perhitungan *lift ratio* dapat dilakukan ketika perhitungan SPADE pada proses sebelumnya menghasilkan kombinasi *itemset*. Berikut merupakan tahapan dalam menghitung *lift ratio*:

1. Data yang dibutuhkan dalam perhitungan *lift ratio* adalah nilai minimum *confidence* dan juga data *itemset* yang telah didapatkan sebelumnya.
2. Menghitung nilai *confidence* dengan rumus persamaan (2-2).
3. Melakukan pengecekan apakah nilai *confidence* lebih dari sama dengan nilai minimum *confidence* yang telah ditetapkan.
4. Melakukan pengecekan apakah nilai *confidence* memenuhi dari nilai minimum *confidence*. Jika iya maka lanjut ke proses berikutnya, jika tidak maka proses membaca data *itemset* berikutnya dan kembali ke proses 3.
5. Menghitung nilai *lift ratio* dengan rumus persamaan (2-4).
6. Menyimpan *itemset* tersebut kedalam *rule*.
7. Jika semua *itemset* telah dihitung maka proses selesai dan menghasilkan kumpulan *rule* dengan nilai *lift ratio*-nya.

Untuk lebih jelasnya mengenai tahap-tahap perhitungan *lift ratio* dapat dilihat pada Gambar 3.9.

3.5 Perhitungan Manual

Secara singkat dalam perhitungan manual akan dilakukan beberapa proses yaitu:

1. Mengambil sampel beberapa data *access logs* untuk digunakan sebagai data transaksi.
2. Mengganti nama *website* dengan indeks yang telah ditentukan untuk mempermudah pembacaan data transaksi dan juga perhitungan.
3. Membentuk data transaksi dengan mengelompokkan IP yang sama menjadi satu SID (*sequence identifier*) dengan EID (*event identifier*) yang disesuaikan dengan urutan waktu transaksi.
4. Melakukan perhitungan SPADE:
 - a. Menemukan *frequent 1-sequence*.
 - b. Menemukan *frequent 2-sequence*.

- c. Menemukan *frequent k-sequence*.
- d. Menghitung nilai *confidence* dan *lift ratio*.

Dari data *access logs* pada *proxy* Universitas Brawijaya diambil beberapa sampel untuk digunakan dalam perhitungan manual. Data *access logs* pada Tabel 3.7 ini merupakan data pada tahun 2011 dengan mengambil data transaksi yang dilakukan oleh enam *IP address*.

Tabel 3.7. Contoh data *access logs* pada *proxy* Universitas Brawijaya.

IP	TANGGAL	JAM	WEBSITE
172.17.104.196	17-02-11	17.30-18.00	blogger.com mediafire.com youtube.com
172.17.104.42	17-02-11	17.34-18.04	google.com kaskus.us webs.com facebook.com ketawa.com youtube.com detik.com
172.17.104.63	17-02-11	18.05-18.35	google.com kaskus.us okezone.com facebook.com nawala.com youtube.com detik.com
172.17.104.84	17-02-11	19.00-19.30	indowebster.com blogspot.com ketawa.com
172.17.104.84	18-02-11	10.08-10.38	hotfile.com digitalhuda.com mediafire.com
172.17.104.190	18-02-11	10.10-10.40	bmw.com google.com mediafire.com
172.17.104.42	20-02-11	01.00-01.30	twitter.com facebook.com digidert.com mediafire.com
172.17.104.196	20-02-11	01.05-01.35	hostmonster.com ziddu.com ebay.com

172.17.104.11	21-02-11	08.15-08.45	kaskus.us gstatic.com detik.com facebook.com sharethis.com
172.17.104.63	22-02-11	13.12-13.42	google.com 21cineplex.com soyjoy.co.id yahoo.co.id indowebster.com
172.17.104.190	25-02-11	07.04-07.34	google.com kaskus.us facebook.com nawala.com
172.17.104.11	26-02-11	09.00-09.30	filestube.com zynga.com twitter.com pixhost.org yahoo.co.id allotrafic.com yieldmanager.com abload.le youtube.com
172.17.104.11	28-02-11	13.04-13.34	kliksaya.com trekearth.com yahoo.co.id

Sumber: PPTI-UB.

Kemudian data pada Tabel 3.7 dikelompokkan berdasarkan *IP address* yang sama dan diberi nama sebagai *sequence identifier* atau SID serta diurutkan berdasarkan waktu atau *event* terjadinya transaksi yang diberi nama *event identifier* atau EID. Contoh pengelompokan *IP address* dan pemberian nama SID serta EID dapat dilihat pada table 3.8. Sedangkan hasil pengelompokan data pada Tabel 3.7 dapat dilihat pada Tabel 3.9.

Tabel 3.8. Contoh pengelompokan IP dan penamaan SID serta EID.

IP	SID	TANGGAL	JAM	EID
172.17.104.196	1	17-02-11	17.30-18.00	1
172.17.104.196	1	20-02-11	01.05-01.35	2
172.17.104.42	2	17-02-11	17.34-18.04	1
172.17.104.63	3	17-02-11	18.05-18.35	1

Sumber: Perancangan.

Tabel 3.9. Data transaksi *access logs*.

SID	EID	WEBSITE
1	1	blogger.com mediafire.com youtube.com
	2	hostmonster.com ziddu.com ebay.com
2	1	google.com kaskus.us webs.com facebook.com ketawa.com youtube.com detik.com
	2	twitter.com digicert.com facebook.com mediafire.com
3	1	google.com kaskus.us okezone.com facebook.com nawala.com youtube.com detik.com
	2	google.com 21cineplex.com soyjoy.co.id yahoo.co.id indowebster.com
4	1	indowebster.com blogspot.com ketawa.com
	2	hotfile.com digitalhuda.com mediafire.com
5	1	bmw.com google.com mediafire.com
	2	google.com kaskus.us facebook.com nawala.com
6	1	kaskus.us gstatic.com

2	facebook.com detik.com sharethis.com
	filestube.com zynga.com twitter.com pixhost.org yahoo.co.id allotrafic.com yieldmanager.com abload.ls youtube.com
3	kliksaya.com trekearth.com yahoo.co.id

Sumber: Perancangan.

Untuk memudahkan dalam pembacaan maka dilakukan pengindeksan nama *website* menjadi angka berdasarkan pengindeksan yang diuraikan pada Tabel 3.10.

Tabel 3.10. Indeks *website*.

Website	Index	Website	Index
blogger.com	1	kaskus.us	18
mediafire.com	2	webs.com	19
hostmonster.cpm	3	facebook.com	20
ziddu.com	4	ketawa.com	21
ebay.com	5	youtube.com	22
google.com	6	twitter.com	23
digicert.com	7	okezone.com	24
nawala.com	8	21cineplex.com	25
soyjoy.co.id	9	indowebster.com	26
blogspot.com	10	ketawa.com	27
hotfile.com	11	digitalhuda.com	28
bmw.com	12	gstatic.com	29
sharethis.com	13	filestube.com	30
zynga.com	14	pixhost.org	31
yahoo.co.id	15	allotrafic.com	32
yieldmanager.com	16	abload.ls	33
kliksaya.com	17	trekearth.com	34

Sumber: Perancangan.

Setelah dilakukan pengindeksan nama *website* maka, didapatkan tabel transaksi *sequence* seperti pada Tabel 3.11.

Tabel 3.11. Data transaksi *sequence* vertical.

SID	EID	WEBSITE
1	1	1, 2, 22
	2	3, 4, 5
2	1	6, 18, 19, 20, 21, 22, 27
	2	23, 7, 20, 2
3	1	6, 18, 24, 20, 8, 22, 27
	2	6, 25, 9, 15, 26
4	1	26, 10, 21
	2	11, 28, 2
5	1	12, 6, 2
	2	6, 18, 20, 8
6	1	18, 29, 27, 20, 13
	2	30, 14, 23, 31, 15, 32, 16, 33, 22
	3	17, 34, 15

Sumber: Perancangan.

Setelah didapatkan tabel transaksi *sequence*, kemudian ditentukan nilai minimal *support* (*min_sup*) yang akan digunakan untuk menentukan *frequent sequences*. Dalam perhitungan manual ini digunakan *min_sup* sebesar 50%. Sehingga *itemset* dianggap *frequent* apabila muncul 50% dari keseluruhan SID, dalam perhitungan ini *min_sup* memiliki nilai 3 dikarenakan jumlah dari SID adalah 6. Untuk menentukan 1-*sequence*, dilakukan *scan* pada data transaksi *sequence*, kemudian dicari *itemset* yang muncul lebih banyak dari *min_sup*. *Itemset* yang memiliki SID-list lebih dari *min_sup*, dalam hal ini 50%, dianggap memenuhi syarat. Apabila suatu *itemset* muncul lebih dari 1 kali dalam 1 SID maka, nilai *support* dari *itemset* tersebut tetap dihitung 1. Contoh, *itemset* 6 muncul pada SID 2 dengan EID 1, SID 3 dengan EID 1 dan 2, dan SID 5 dengan

EID 1 dan 2 maka, nilai *support* dari *itemset* 6 adalah 3, dikarenakan *itemset* 6 muncul pada 3 SID meskipun muncul di 5 EID. Tabel 3.12 merupakan daftar *frequent 1-sequence* setelah dilakukan proses *scan* pada data transaksi.

Tabel 3.12. *Frequent 1-sequence.*

WEBSITE					
2		6		18	
SID	EID	SID	EID	SID	EID
1	1	2	1	2	1
2	2	3	1	3	1
4	2	3	2	5	2
5	1	4	1	6	1
		4	2		
WEBSITE					
20		22		27	
SID	EID	SID	EID	SID	EID
2	1	1	1	2	1
2	2	2	1	3	1
3	1	3	1	6	1
5	2	6	2		
6	1				

Sumber: Perancangan.

Daftar *frequent 1-sequence* tersebut digunakan untuk membentuk daftar *frequent 2-sequence* dengan cara menggabungkan antar satu *item* dengan *item* lainnya termasuk dengan *item* itu sendiri. Kemudian gabungan kedua *item* atau *itemset* tersebut dihitung frekuensinya untuk mengetahui apakah *itemset* tersebut memenuhi nilai *min_sup*. Contoh, *item* 6 dikombinasikan dengan *item* 8 maka, kemungkinan hasil penggabungan tersebut adalah 6 , 8 atau 8 , 6 yang memiliki arti sama yaitu *item* 6 dan *item* 8 berada pada SID dan EID yang sama. Kemungkinan hasil kombinasi lainnya adalah 6 -> 8 yang memiliki arti bahwa *item*

8 muncul pada EID setelah *item* 6 muncul pada SID yang sama dan 8 -> 6 yang berarti bahwa *item* 6 muncul pada EID setelah *item* 8 muncul pada SID yang sama. Perhitungan nilai *support* pada tiap *itemset* dilihat dari jumlah SID munculnya kombinasi tersebut. Hasil dari kombinasi *item-item* pada *frequent 1-sequence* yang memenuhi nilai *min_sup* terdapat pada Tabel 3.13.

Tabel 3.13. *Frequent 2-sequence.*

WEBSITE					
6 , 18		6 , 20		18 , 20	
SID	EID	SID	EID	SID	EID
2	1	2	1	2	1
3	1	3	1	3	1
5	2	5	2	5	2
				6	1
WEBSITE					
18 , 27		20 , 27			
SID	EID	SID	EID		
2	1	2	1		
3	1	3	1		
6	1	6	1		

Sumber: Perancangan.

Untuk mendapatkan *frequent sequence* berikutnya maka, daftar *frequent 2-sequences* tersebut digunakan untuk membentuk daftar *frequent 3-sequences*. *Itemset-itemset* pada *2-sequences* digabungkan antar *itemset*-nya untuk menemukan kombinasi-kombinasi yang membentuk *itemset* baru yang memiliki susunan 3 *item* atau *3-sequences*. Terdapat tiga kemungkinan dalam pembentukan *3-sequences* yaitu:

1. Jika dilakukan penggabungan *itemset* A,B dengan A,C maka, kemungkinan hasilnya adalah A,B,C.

2. Jika dilakukan penggabungan *itemset* A,B dengan $A \rightarrow C$ maka, kemungkinan hasilnya adalah $A,B \rightarrow C$.
3. Jika dilakukan penggabungan *itemset* $A \rightarrow B$ dengan $A \rightarrow C$ maka, kemungkinan hasilnya ada 3 yaitu $A \rightarrow B,C$ atau $A \rightarrow B \rightarrow C$ atau $A \rightarrow C \rightarrow B$.

Contoh, *itemset* 18,27 akan dikombinasikan dengan 20,27 maka, kemungkinan dari hasil kombinasinya adalah 18,20,27.

Hasil dari pencarian 3-*sequence* dapat dilihat pada Tabel 3.14 dimana 3-*sequence* merupakan *sequence* terakhir dikarenakan kombinasi dari 3-*sequence* yang membentuk 4-*sequence* tidak memenuhi dari minimum *support* yaitu kombinasi 6,18,20,27 yang hanya memiliki nilai *support* sebesar 2.

Tabel 3.14. Frequent 3-*sequence*.

WEBSITE			
6 , 18 , 20		18 , 20 , 27	
SID	EID	SID	EID
2	1	2	1
3	1	3	1
5	2	6	1

Sumber: Perancangan.

Tabel 3.15. Lift ratio rule.

Rule	Confidence	Benchmark Confidence	Lift Ratio
6 , 18	1	0,6667	1,5
6 , 20	1	0,6667	1,5
18 , 20	1	0,6667	1,5
18 , 27	0,75	0,5	1,5
20 , 27	0,75	0,5	1,5
6 , 18 , 20	1	0,6667	1,5
18 , 20 , 27	0,75	0,5	1,5

Sumber: Perancangan.

Setelah menemukan *k-sequence* dalam hal ini *3-sequence* maka, selanjutnya dilakukan perhitungan *lift ratio* dari setiap *rule*. Hasil perhitungan tersebut dapat dilihat pada Tabel 3.15. Minimum *confidence* pada perhitungan kali ini bernilai 0,6.

Suatu *rule* dapat dikatakan kuat jika nilai *lift ratio* dari *rule* tersebut lebih dari 1. Dari Tabel 3.15 maka didapatkan bahwa semua *rule* yang dihasilkan kuat dengan nilai *lift ratio* sebesar 1,5. Terdapat *rule* 6 , 18 yang memiliki arti jika mengakses google.com maka mengakses nawala.com, dan 6, 18, 20 jika mengakses google.com dan nawala.com maka mengakses facebook.com.

3.6 Perancangan Antarmuka (User Interface)

Rancangan *user interface* dapat dilihat pada gambar 3.10, 3.11, 3.12 , 3.13 dan 3.14.

The image shows a user interface window titled "PREPROCESSING DATA". Inside the window, there are three elements arranged vertically: a rectangular text input field labeled "A", a rectangular button labeled "B", and another rectangular button labeled "C".

Gambar 3.10. Rancangan *User Interface* halaman *preprocessing* data.

Gambar 3.10 merupakan rancangan halaman *preprocessing* data yang dijalankan untuk memproses data *access logs* dalam bentuk txt menjadi data yang disimpan pada database. Berikut bagian-bagian halaman tersebut:

- A. Tombol untuk memilih file .txt.
- B. Tombol untuk memproses file untuk disimpan pada database.
- C. Tombol untuk memulai ulang halaman *preprocessing* data.

Pada gambar 3.11 merupakan rancangan halaman utama dan pertama kali muncul ketika aplikasi pencarian pola perilaku pengguna *internet* dijalankan dengan keterangan sebagai berikut:

- A. *Combobox* untuk memilih nama fakultas.
- B. *Combobox* untuk memilih kategori waktu.
- C. *Textfield* untuk memasukkan nilai batas waktu.
- D. Tombol untuk membentuk data transaksi.
- E. Tombol untuk menampilkan kode-kode *website*.
- F. Tombol untuk keluar dari aplikasi.
- G. Tempat menampilkan hasil pembentukan data transaksi.

PENCARIAN POLA PERILAKU PENGGUNA INTERNET

Nama Fakultas

Kategori Waktu

Batas Waktu

G

Gambar 3.11. Rancangan *User Interface* halaman utama.

Pada gambar 3.12 merupakan rancangan halaman utama setelah data transaksi berhasil dibentuk. Berikut merupakan keterangan tiap-tiap bagiannya:

- A. Tombol untuk menampilkan kode-kode *website*.
- B. *Radio Button* untuk memilih jenis minimum support yang akan dimasukan, dalam bentuk frequent atau dalam bentuk prosentase.
- C. *Textfield* untuk memasukkan minimum support dalam bentuk prosentase.

- D. *Textfield* untuk memasukkan minimum support dalam bentuk frequent.
- E. *Textfield* untuk memasukkan minimum confidence dalam bentuk prosentase.
- F. Tombol untuk memulai kembali aplikasi mulai awal.
- G. Tombol untuk melakukan perhitungan SPADE.
- H. Tombol untuk keluar dari aplikasi.
- I. Tempat menampilkan data transaksi.

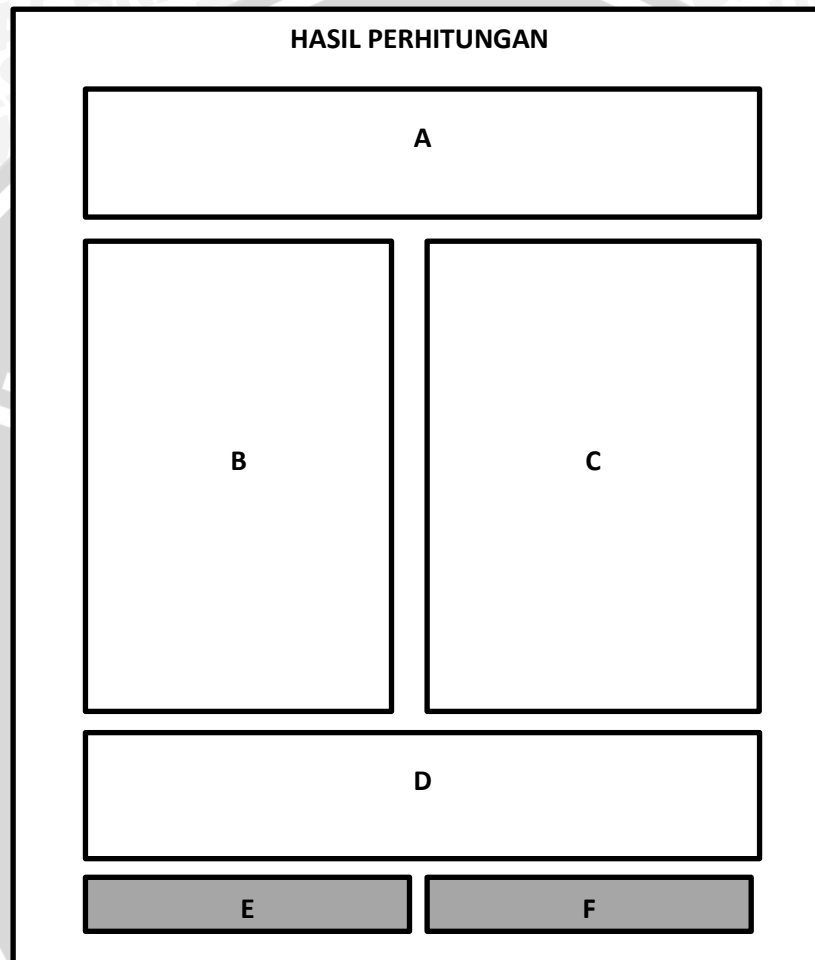
The image shows a user interface design for a web application titled "PENCARIAN POLA PERILAKU PENGGUNA INTERNET". The interface is contained within a rectangular frame. At the top center is the title. Below the title, there are several interactive elements: a button labeled 'A' at the top; a label "Minimum Support" followed by two radio button options, 'B' (Prosentase) and 'C' (Frequent), with corresponding text input fields 'D' and 'E'; a label "Minimum Confidence" followed by a text input field 'E'; a large empty rectangular area labeled 'I' on the right side; and at the bottom, three buttons labeled 'F', 'G', and 'H' arranged horizontally.

Gambar 3.12. Rancangan *User Interface* halaman utama setelah data transaksi terbentuk.

Pada gambar 3.13 merupakan rancangan halaman hasil perhitungan SPADE. Berikut merupakan keterangan tiap-tiap bagiannya:

- A. Status atau keterangan dari masukan awal untuk melakukan perhitungan SPADE.
- B. Tempat untuk menampilkan *itemset sequence*.
- C. Tempat untuk menampilkan rule yang dihasilkan dari perhitungan SPADE.

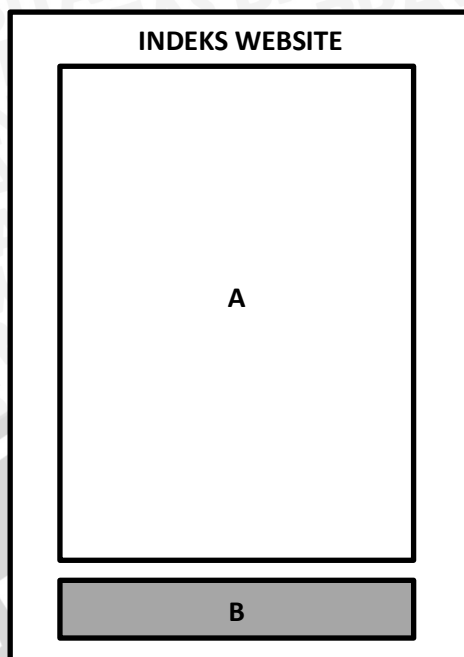
- D. Tempat untuk menampilkan rangkuman atau kesimpulan dari rule yang telah dihasilkan.
- E. Tombol untuk menampilkan kode-kode *website*.
- F. Tombol untuk keluar dari jendela hasil perhitungan.



Gambar 3.13. Rancangan *User Interface* halaman hasil perhitungan.

Pada gambar 3.14 merupakan rancangan halaman indeks *website*. Berikut merupakan keterangan tiap-tiap bagiannya:

- A. Tempat untuk menampilkan *website* dan kodennya.
- B. Tombol untuk keluar dari jendela *index website*.



Gambar 3.14. *User Interface* halaman *index website*.

3.7 Perancangan Pengujian

Pengujian pada sistem pencarian pola perilaku pengguna internet ini dengan melakukan evaluasi terhadap analisa yang dihasilkan oleh sistem. Tujuan dari pengujian ini adalah untuk mengetahui pengaruh nilai minimum *support* dan minimum *confidence* terhadap jumlah rule yang dihasilkan serta mengetahui besaran *lift ratio* pada rule yang dihasilkan.

Pengujian untuk mengetahui pengaruh nilai minimum *support* dan minimum *confidence* dilakukan pada fakultas yang memiliki transaksi terbesar dari ketujuh fakultas yang ada sesuai dengan tabel 3.2 dengan 3 kategori waktu yaitu jam kerja, jam istirahat, dan diluar jam kerja. Pembatasan waktu transaksi di kategorikan mulai 10 menit, 20 menit hingga 30 menit. Pada pengujian minimum *support* dilakukan dengan merubah nilai minimum *support* tanpa merubah nilai minimum *confidence* di setiap pengujiannya. Begitu pula sebaliknya pada pengujian nilai minimum *confidence*. Sedangkan pengujian untuk mengetahui besaran *lift ratio* pada rule yang dihasilkan dilakukan pada seluruh fakultas pada 3 kategori waktu yaitu jam kerja, jam istirahat, dan diluar jam kerja.

BAB IV

IMPLEMENTASI

4.1 Lingkungan Implementasi

Lingkungan implementasi merupakan keadaan atau kondisi yang digunakan dalam melakukan eksekusi dari perancangan yang telah dibentuk sebelumnya. Lingkungan implementasi pada penelitian ini dibagi menjadi dua yaitu lingkungan implementasi perangkat keras dan lingkungan implementasi perangkat lunak.

4.1.1 Lingkungan Perangkat Keras

Lingkungan perangkat keras yang digunakan pada penelitian ini adalah sebagai berikut:

1. Processor Intel® Core™ i3 2,5 GHz
2. Memory 6 GB
3. Harddisk dengan kapasitas 450 GB
4. Monitor 12"
5. Keyboard
6. Mouse

4.1.2 Lingkungan Perangkat Lunak

Lingkungan perangkat keras yang digunakan pada penelitian ini adalah sebagai berikut:

1. Windows 7 Professional 64-bit , digunakan sebagai *operating system* pada pembuatan hingga pengujian sistem pencarian pola perilaku pengguna internet.
2. Netbeans IDE 7.4 , merupakan sebuah *tool* pemrograman yang digunakan untuk melakukan pemrograman sistem pola pencarian perilaku pengguna internet, dalam hal ini bahasa pemrograman yang digunakan adalah Java.

3. XAMPP version 1.8.1 , digunakan sebagai *tool* untuk menyediakan penyimpanan pada *database* atau *server*.
4. Google Chrome version 33.0 , digunakan untuk mengakses *database* penyimpanan sistem pencarian pola perilaku pengguna internet.

4.2 Implementasi Program

Implementasi program merupakan implementasi setelah dilakukannya perancangan dan analisa sistem pada bab sebelumnya. Pada subbab ini akan dijelaskan implementasi proses-proses yang dijalankan oleh sistem.

4.2.1 Implementasi *Preprocessing Data Access Logs*

Preprocessing data access logs merupakan proses mempersiapkan atau filterisasi pada data *access logs* untuk mengumpulkan informasi-informasi yang dibutuhkan oleh sistem pencarian pola perilaku pengguna internet. Implementasi *preprocessing data* dapat dilihat pada sourcecode 4.1.

```
if ("text/html".equals(type)) {  
    url = filter.UrlFilter(url);  
    url = filter.DomainFilter(url);  
    indeks = filter.IndeksFilter(url);  
    System.out.println(url+"-"+indeks);  
    time = date.getDate(id);  
    System.out.println(time);  
    unix = date.getUnix(id);  
    System.out.println(unix);  
    kategori = date.getKategori(id);  
    System.out.println(kategori);  
    if("ERROR".equals(url)) {  
    } else {  
        total++;  
        save.SaveData(id, time, ip, url, kategori, unix);  
        save.SaveTrans(id, time, ip, indeks, kategori, unix);  
        System.out.println(total);  
    }  
}
```

}

Sourcecode 4.1 *Preprocessing data Access Logs.*

Inti dari proses yang dilakukan pada *preprocessing* data adalah menyaring berbagai jenis informasi yang dibutuhkan seperti waktu, *url*, *MIMEtype* serta *IP address*.

4.2.2 Implementasi Pembentukan Data Transaksi

Proses pembentukan data transaksi merupakan proses yang bertujuan untuk membentuk data transaksi yang digunakan sebagai pedoman dalam perhitungan SPADE. Pada implementasi proses ini, data transaksi disimpan dalam bentuk *linked list*. Implementasi pembentukan data transaksi dapat dilihat pada *sourcecode 4.2*.

```
while (hasilA.next()) {
    mPlus = mPlusTemp;
    mAwal = mAwalTemp;
    mMaks = mMaksTemp;
    mAkhir = mAwal + mPlus;
    sid++;
    eid = 0;
    while (mAkhir <= mMaks) {
        statementB = Koneksi.getConnection().createStatement();
        hasilB = statementB.executeQuery("select * from access where ip=" +
            hasilA.getString("ip").toString() + " and unix>=" + mAwal + " and unix<" + mAkhir + " ");
        mAwal = mAkhir;
        mAkhir = mAwal + mPlus;
        eid++;
        if ((mAkhir == mMaks) && (cekKategori == false)) {
            mAwal = mAwal2Temp;
            mMaks = mMaks2Temp;
            mAkhir = mAwal + mPlus;
            cekKategori = true;
        }
        while (hasilB.next()) {
```

```

url = hasilB.getString("url").toString();
while ((temp2 != null) && (cekNode == true)) {
    if ((temp2.getSID() == sid) && (temp2.getEID() == eid) &&
(temp2.getURL().equals(url))) {
        cekNode = false;
        cekUrl = true;
    }
    temp2 = temp2.getNext();
}
cekNode = true;
if (cekUrl == false) {
    node = new DataTransaksi();
    node.DataTransaksi(url, sid, eid, temp);
    temp = node;
    if (second != null) {
        second.setPrev(temp);
    }
    if (first == null) {
        first = temp;
    }
    second = temp;
}
cekUrl = false;
temp2 = temp;
}
statementB.close();
}
}
statementA.close();
this.data = first;
this.sumData = sid;
}

```

Sourcecode 4.2 Pembentukan Data Transaksi.

4.2.3 Implementasi Perhitungan SPADE

Implementasi perhitungan SPADE merupakan proses-proses yang dilakukan sistem dalam melakukan perhitungan SPADE. Pada proses ini dibagi

menjadi 4 proses yaitu proses *frequent 1-sequence*, proses *frequent 2-sequence*, proses *frequent 3-sequence* dan proses perhitungan lift ratio.

4.2.3.1 Frequent 1-Sequence

Proses *frequent 1-sequence* merupakan proses yang bertujuan untuk menemukan *itemset* yang terdiri dari 1 *item* yang memiliki nilai *frequent* yang melebihi nilai minimum *support*. Proses *frequent 1-sequence* dapat dilihat pada *sourcecode* 4.3.

```
statement = Koneksi.getConnection().createStatement();
hasil = statement.executeQuery("select indeks from indexweb");
while (hasil.next()) {
    indeks = hasil.getString("indeks").toString();
    total = 0;
    sid = 0;
    while (data != null) {
        if ((indeks.equals(data.getUrl())) && (sid != data.getSID())) {
            total++;
            sid = data.getSID();
        }
        data = data.getPrev();
    }
    if (total >= minSup) {
        itemset = new Itemset();
        items = new Items();
        items.setItem(indeks);
        items.setRelasi("");
        items.setNext(tempItems);
        itemset.setAntacendant(items);
        itemset.setFrequent(total);
        itemset.setSequence(1);
        itemset.setStatus("");
        itemset.setNext(tempItemset);
        tempItemset = itemset;
        if (prev != null) {
            prev.setPrev(tempItemset);
        }
        prev = tempItemset;
    }
}
```

```

data = SaveData;
}
statement.close();
this.Itemset = tempItemset;
temp = tempItemset;

```

Sourcecode 4.3 *Frequent 1-Sequence.*

4.2.3.2 *Frequent 2-Sequence*

Proses *frequent 2-sequence* merupakan proses yang bertujuan untuk menemukan kombinasi dari *itemset frequent 1-sequence* yang memiliki nilai *frequent* yang melebihi nilai minimum *support*. Proses *frequent 2-sequence* dapat dilihat pada *sourcecode 4.4*.

```

while (itemsetA != null) {
    itemsetB = itemset;
    while (itemsetB != null) {
        if (!itemsetA.getItems().equals(itemsetB.getItems())) {
            for (int i = 1; i <= 2; i++) {
                baru = new Itemset();
                item = new Items();
                item.setItem(itemsetA.getItems());
                item.setRelasi("");
                item.setNext(null);
                baru.setAntacendant(item);
                item = new Items();
                item.setItem(itemsetB.getItems());
                if (i == 1) {
                    item.setRelasi(",");
                } else {
                    item.setRelasi("->");
                }
                item.setNext(null);
                baru.setConsequant(item);
                baru.setSequence(2);
                if (i == 1) {

```

```

        baru.setStatus("cek");
    } else {
        baru.setStatus("");
    }
    baru.setNext(temp);
    temp = baru;
    if (prev != null) {
        prev.setPrev(temp);
    }
    prev = temp;
}
}
itemsetB = itemsetB.getNext();
}
itemsetA = itemsetA.getNext();
}
}

```

Sourcecode 4.4 *Frequent 2-Sequence.*

4.2.3.3 Frequent k -Sequence

Proses *frequent k-sequence* merupakan proses yang bertujuan untuk menemukan kombinasi dari *itemset frequent 2-sequence* yang memiliki nilai *frequent* yang melebihi nilai minimum *support*. Dan berulang melakukan kombinasi dari $k-1$ sequence hingga tidak ditemukan kombinasi lagi. Proses pembentukan *k-sequence* dapat dilihat pada sourcecode 4.5.

```

while (kCek == true) {
    head = this.Itemset;
    kombinasi = null;
    itemsetA = this.Itemset;
    System.out.println("k:"+k);
    while ((itemsetA != null) && (itemsetA.getSequence() == (k - 1))) {
        itemsetB = itemsetA.getNext();
        while ((itemsetB != null) && (itemsetB.getSequence() == (k - 1))) {
            newAntac = newConseq = tempo = null;
            cek = false;
            temAntac = temConseq = tempo2 = null;

```



```

temAntac2 = temConseq2 = tempo3 = null;
if (itemsetA.getAntac().equals(itemsetB.getAntac())) {
    itemA = itemsetA.getConsequant();
    itemB = itemsetB.getConsequant();
    if (("->".equals(itemA.getRelasi())) && ("->".equals(itemB.getRelasi())) {
        cek = true;
    }
    itemA = itemsetA.getAntacendant();
    while (itemA != null) {
        node = new Items();
        node.setItems(itemA.getRelasi(), itemA.getItem());
        if (newAntac == null) {
            newAntac = node;
        } else {
            tempo.setNext(node);
        } tempo = node;
        if (cek == true) {
            node = new Items();
            node.setItems(itemA.getRelasi(), itemA.getItem());
            if (temAntac == null) {
                temAntac = node;
            } else {
                tempo2.setNext(node);
            } tempo2 = node;
            node = new Items();
            node.setItems(itemA.getRelasi(), itemA.getItem());
            if (temAntac2 == null) {
                temAntac2 = node;
            } else {
                tempo3.setNext(node);
            } tempo3 = node;
        } itemA = itemA.getNext();
    } itemA = itemsetA.getConsequant();
    itemB = itemsetB.getConsequant();
    if (",".equals(itemA.getRelasi())) {
        node = new Items();
        node.setItems(itemA.getRelasi(), itemA.getItem());
        if (newAntac == null) {

```

```

        newAntac = node;
    } else {
        tempo.setNext(node);
    }
    tempo = node;
    node = new Items();
    node.setItems(itemB.getRelasi(), itemB.getItem(), newConseq);
    newConseq = node;
}

```

Sourcecode 4.5 *Frequent k-Sequence.*

4.2.4 Implementasi Perhitungan *Lift Ratio*

Proses perhitungan *lift ratio* merupakan proses yang dijalankan untuk mendapatkan nilai *lift ratio* dari setiap *rule* yang terbentuk. Proses *lift ratio* ini dilakukan setelah *frequent k-sequence* terbentuk. Proses perhitungan *lift ratio* dapat dilihat pada *sourcecode 4.6*.

```

public void setBenchMark(Itemset itemset, long sumSID) {
    Itemset itemA, itemB;
    long conseq;
    double Conseq, Sum, BC, lift;
    String CONSEQ;
    System.out.println("benchmark");
    itemA = itemset;
    Sum = sumSID;
    int decimalPlace = 3;
    while (itemA != null) {
        Conseq = 0;
        if (itemA.getSequence() > 1) {
            CONSEQ = itemA.getConseq();
            itemB = itemset;
            while (itemB != null) {
                if (itemB.getItems().equals(CONSEQ)) {
                    conseq = itemB.getFrequent();
                    Conseq = (double) conseq;
                }
            }
        }
    }
}

```

```

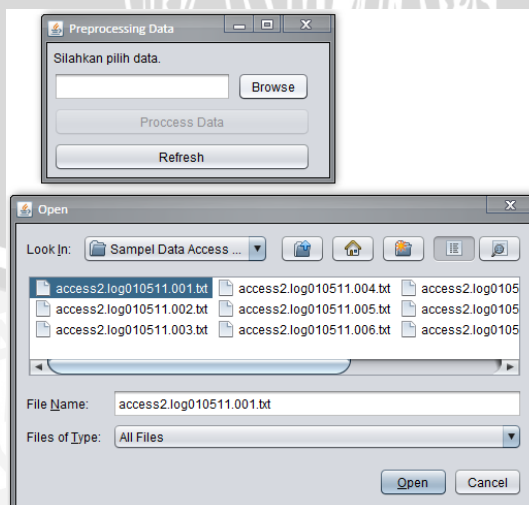
    }
    itemB = itemB.getNext();
}
BC = Conseq / Sum;
BigDecimal bd = new BigDecimal(BC);
bd = bd.setScale(decimalPlace, BigDecimal.ROUND_UP);
BC = bd.doubleValue();
itemA.setBenchmark(BC);
lift = (itemA.getConfidence() / itemA.getBenchmark());
System.out.println("lift" + lift);
BigDecimal br = new BigDecimal(lift);
br = br.setScale(decimalPlace, BigDecimal.ROUND_UP);
lift = br.doubleValue();
itemA.setLifratio(lift);
}
itemA = itemA.getNext();
}
}

```

Sourcecode 4.6 Lift Ratio.

4.3 Implementasi Antarmuka

Pada gambar 4.7 merupakan halaman untuk melakukan *preprocessing* data. *User* diminta memasukkan *file data access logs* yang selanjutnya data tersebut akan disaring dan disimpan pada *database*.



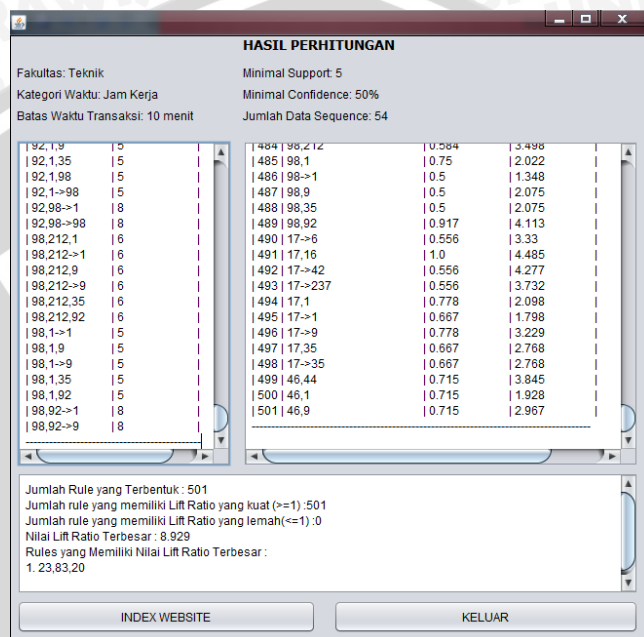
Gambar 4.7. Halaman *preprocessing* data.

Gambar 4.8 merupakan halaman utama pada sistem pencarian pola perilaku pengguna *internet*. *User* diminta untuk memilih nama fakultas, kategori waktu dan juga batasan waktu untuk membuat data transaksi. Setelah data transaksi terbentuk, *user* diminta untuk memasukkan batasan nilai minimum *support* dan nilai minimum *confidence* untuk mendapatkan hasil perhitungan seperti pada gambar 4.9.

Gambar 4.8. Halaman utama.

Gambar 4.9. Halaman utama setelah didapatkan data transaksi.

Gambar 4.10 merupakan halaman hasil dari perhitungan yang ditampilkan setelah *user* menekan tombol “Perhitungan”. Pada halaman hasil terdapat tombol “Index Web” yang digunakan untuk melihat alamat *website* yang telah dikodekan sebelumnya seperti pada gambar 4.11.

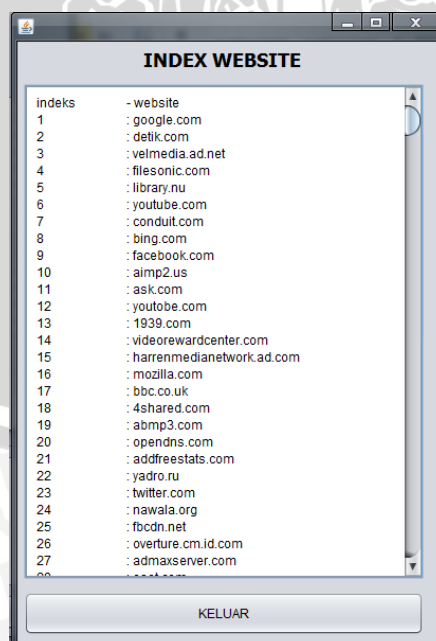


Fakultas: Teknik		Minimal Support: 5	
Kategori Waktu: Jam Kerja		Minimal Confidence: 50%	
Batas Waktu Transaksi: 10 menit		Jumlah Data Sequence: 54	
192,1,9	15	1484 98,212	0,584
192,1,35	15	1485 98,1	0,75
192,1,98	15	1486 98~1	0,5
192,1~98	15	1487 98,9	0,5
192,98~1	18	1488 98,35	0,5
192,98~98	18	1489 98,92	0,917
198,212,1	16	1490 17~6	0,556
198,212~1	16	1491 17,16	1,0
198,212,9	16	1492 17~42	0,556
198,212~9	16	1493 17~237	0,556
198,212,35	16	1494 17,1	0,778
198,212,92	16	1495 17~1	0,667
198,1~1	15	1496 17~9	0,778
198,1,9	15	1497 17,35	0,667
198,1~9	15	1498 17~35	0,667
198,1,35	15	1499 46,44	0,715
198,1,92	15	1500 46,1	0,715
198,92~1	18	1501 46,9	0,715
198,92~9	18		

Jumlah Rule yang Terbentuk: 501
 Jumlah rule yang memiliki Lift Ratio yang kuat (≥ 1): 501
 Jumlah rule yang memiliki Lift Ratio yang lemah (≤ 1): 0
 Nilai Lift Ratio Terbesar: 8.929
 Rules yang Memiliki Nilai Lift Ratio Terbesar:
 1. 23.83.20

INDEX WEBSITE KELUAR

Gambar 4.10. Halaman hasil.



indeks	- website
1	: google.com
2	: detik.com
3	: velmedia.ad.net
4	: filesonic.com
5	: library.nu
6	: youtube.com
7	: conduit.com
8	: bing.com
9	: facebook.com
10	: aimp2.us
11	: ask.com
12	: youtube.com
13	: 1939.com
14	: videorewardcenter.com
15	: harrenmedianetwork.ad.com
16	: mozilla.com
17	: bbc.co.uk
18	: 4shared.com
19	: abmp3.com
20	: opendns.com
21	: addfreestats.com
22	: yadro.ru
23	: twitter.com
24	: nawala.org
25	: fbcdn.net
26	: overture.cm.id.com
27	: admaxserver.com

KELUAR

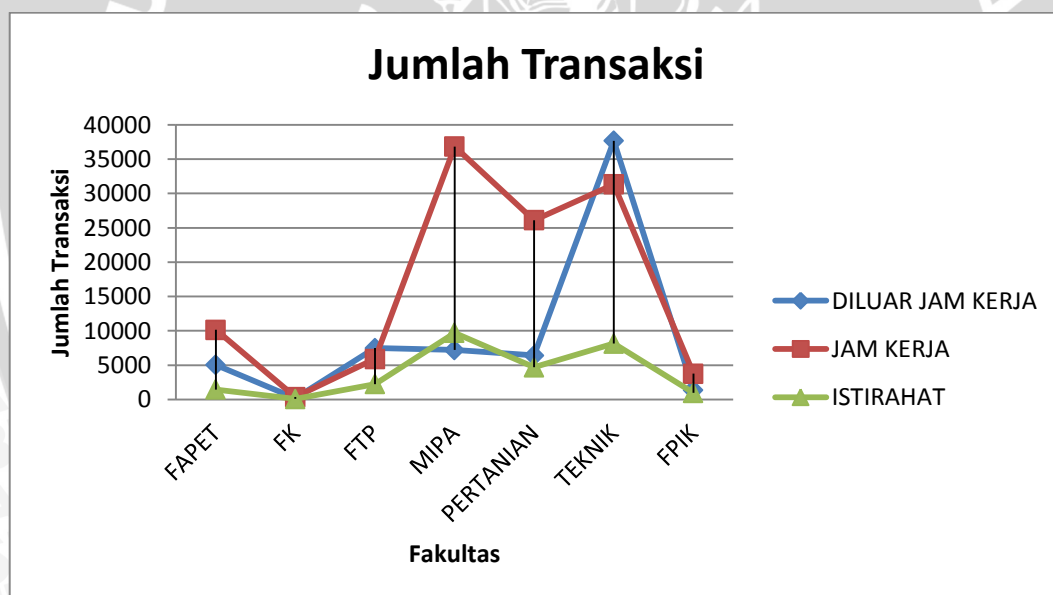
Gambar 4.11. Halaman *index web*.

BAB V

PENGUJIAN DAN ANALISIS

5.1 Hasil Pengujian

Pengujian dilaksanakan sesuai dengan skenario pengujian yang telah dirancang sebelumnya untuk mendapatkan hasil yang sesuai dengan tujuan penelitian. Dibutuhkan sampel data dari salah satu fakultas untuk melakukan pengujian berdasarkan minimum *support* dan minimum *confidence*. Fakultas yang dijadikan sampel pada pengujian tersebut merupakan fakultas yang memiliki data transaksi terbanyak. Jumlah data transaksi didapatkan dengan menghitung jumlah transaksi pada setiap fakultas.



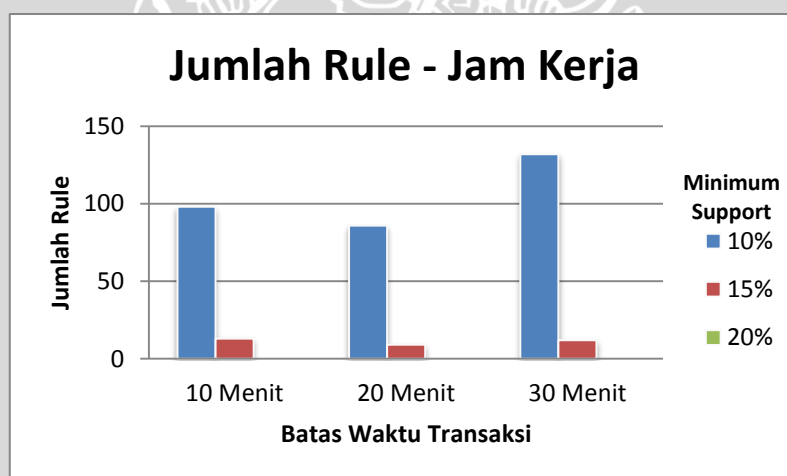
Gambar 5.1 Grafik Jumlah Transaksi.

Pada gambar 5.1 didapatkan grafik jumlah transaksi dari ketujuh fakultas yang ada. Jika dilakukan penjumlahan maka, Fakultas Teknik merupakan fakultas dengan jumlah transaksi terbanyak dengan jumlah 77110 transaksi. Oleh karena itu Fakultas Teknik dijadikan sampel untuk pengujian minimum *support* dan minimum *confidence*.

5.1.1 Hasil Pengujian Berdasarkan Minimum Support

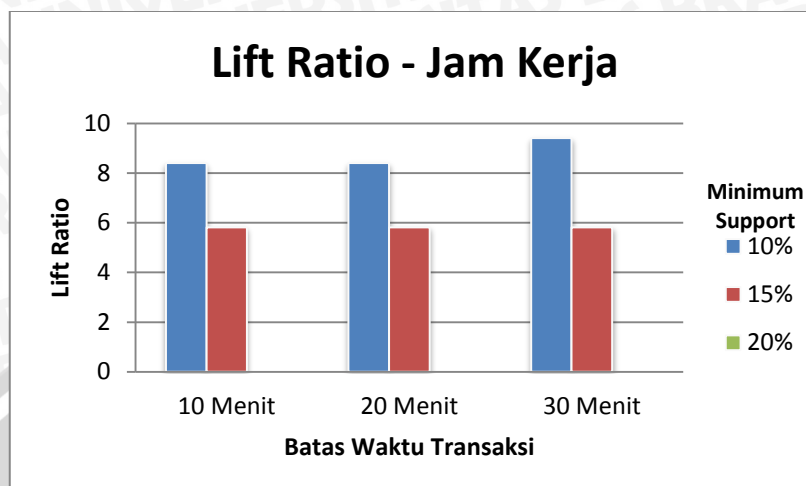
Uji coba pada pengujian berdasarkan minimum *support* menggunakan nilai minimum *confidence* yang tetap yaitu sebesar 50%. Sedangkan nilai minimum *support* diambil secara *random*. Uji coba pertama dilakukan pada Fakultas Teknik dengan kategori waktu pada saat jam kerja.

Gambar 5.2 merupakan hasil pengujian pertama pada jumlah *rule* yang dihasilkan. Jumlah *rule* terbanyak sebesar 132 *rule* dihasilkan ketika nilai minimum *support* sebesar 10% dengan batas waktu transaksi 30 menit. Jumlah *rule* pada minimum *support* 10% lebih banyak dari jumlah *rule* dengan minimum *support* 15%. Sedangkan ketika nilai minimum *support* 20% tidak terdapat *rule* yang dihasilkan. Pada batas waktu 20 menit, jumlah *rule* yang dihasilkan merupakan jumlah *rule* paling sedikit dibandingkan pada batas waktu yang lain. Hal ini dikarenakan jangka waktu akses pada suatu halaman *website* yang lama oleh pengguna *internet* pada saat jam kerja.



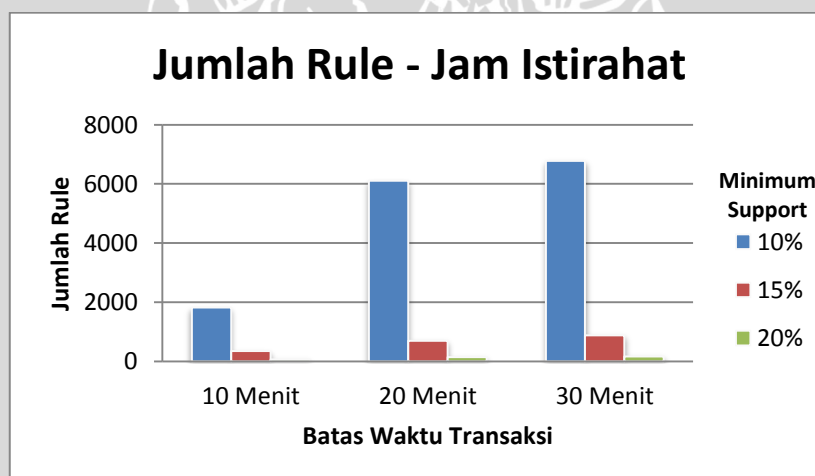
Gambar 5.2 Grafik Minimum Support Jumlah Rule - Jam Kerja.

Gambar 5.3 merupakan hasil pengujian pertama pada besar *lift ratio* yang dihasilkan. *Lift ratio* terbesar dengan nilai 9,409 didapatkan ketika batas waktu transaksi 30 menit dengan minimum *support* 10%. *Lift ratio* pada minimum support 10% lebih besar dari *lift ratio* dengan minimum support 15%. Sedangkan pada minimum support 20% nilai *lift ratio* yang dihasilkan 0, dikarenakan tidak terdapat *rule* yang dihasilkan.



Gambar 5.3 Grafik Minimum Support Lift Ratio- Jam Kerja.

Uji coba kedua dilakukan pada Fakultas Teknik dengan kategori waktu pada saat jam istirahat.

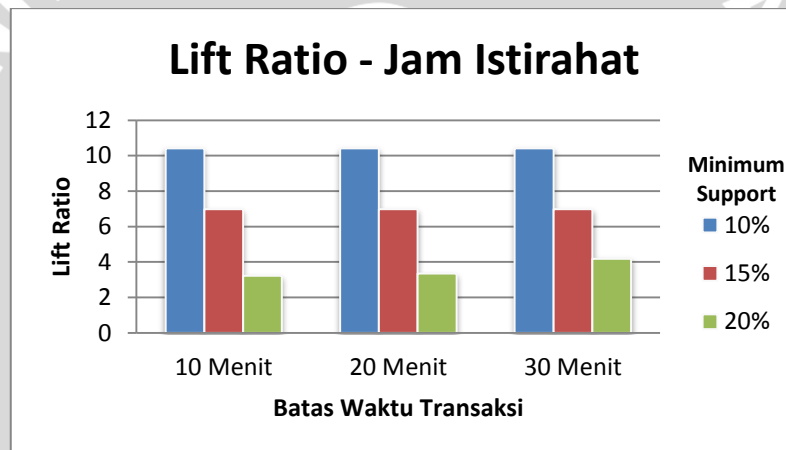


Gambar 5.4 Grafik Minimum Support Jumlah Rule - Jam Istirahat.

Gambar 5.4 merupakan hasil pengujian kedua pada jumlah *rule* yang dihasilkan. Jumlah *rule* terbanyak sebesar 6779 *rule* dihasilkan ketika nilai minimum *support* sebesar 10% dengan batas waktu transaksi 30 menit. Jumlah *rule* pada minimum *support* 10% lebih banyak dari jumlah *rule* dengan minimum *support* 15% dan juga lebih banyak dari jumlah *rule* dengan minimum *support*

20%. Dilihat dari batas waktu transaksi, semakin lama batas waktu yang ditentukan, jumlah *rule* yang dihasilkan semakin banyak. Hal ini dikarenakan pada jam istirahat jangka waktu akses terhadap suatu halaman *website* yang sebentar atau banyaknya variasi halaman *website* yang diakses oleh pengguna *internet* pada jam istirahat.

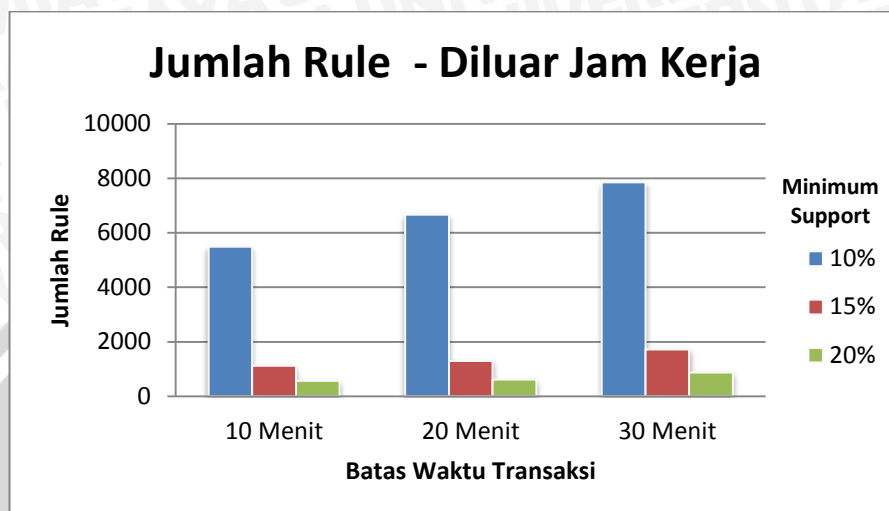
Gambar 5.5 merupakan hasil pengujian kedua pada besar *lift ratio* yang dihasilkan. *Lift ratio* terbesar dengan nilai 10,417 didapatkan ketika minimum *support* 10%. *Lift ratio* pada minimum *support* 10% lebih besar dari *lift ratio* dengan minimum *support* 15% dan juga lebih besar dari *lift ratio* dengan minimum *support* 20%.



Gambar 5.5 Grafik Minimum *Support Lift Ratio*- Jam Istirahat.

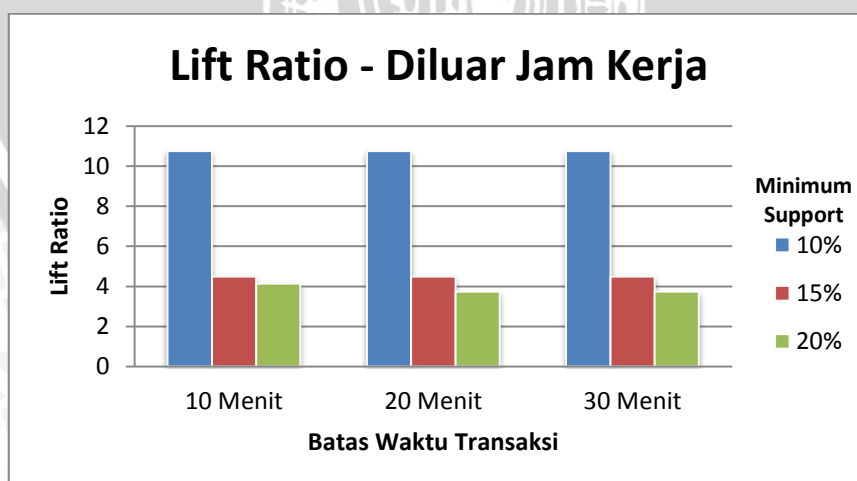
Uji coba ketiga dilakukan pada Fakultas Teknik dengan kategori waktu pada saat jam diluar jam kerja. Gambar 5.6 merupakan hasil pengujian ketiga pada jumlah *rule* yang dihasilkan. Jumlah *rule* terbanyak sebesar 7848 *rule* dihasilkan ketika nilai minimum *support* sebesar 10% dengan batas waktu transaksi 30 menit. Jumlah *rule* pada minimum *support* 10% lebih banyak dari jumlah *rule* dengan minimum *support* 15% dan juga lebih banyak dari jumlah *rule* dengan minimum *support* 20%. Dilihat dari batas waktu transaksi, semakin lama batas waktu yang ditentukan, jumlah *rule* yang dihasilkan semakin banyak. Hal ini dikarenakan pada kategori waktu diluar jam kerja jangka waktu akses terhadap

suatu halaman *website* yang sebentar atau banyaknya variasi halaman *website* yang diakses oleh pengguna *internet* pada jam istirahat.



Gambar 5.6 Grafik Minimum Support Jumlah Rule – Diluar Jam Kerja.

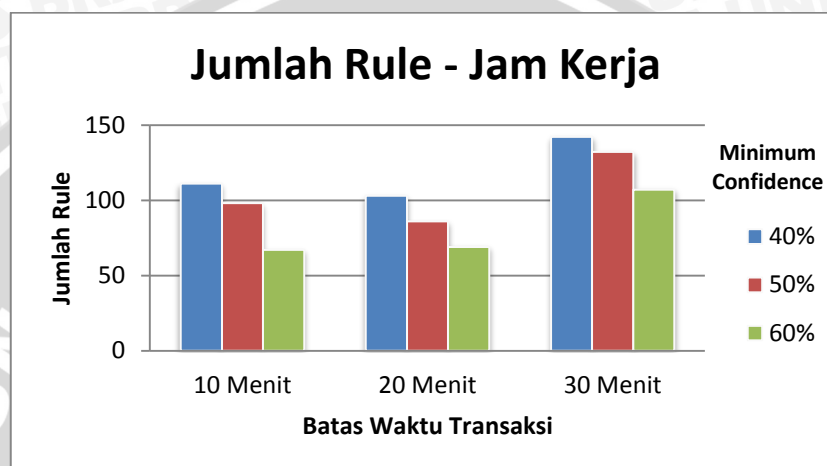
Gambar 5.7 merupakan hasil pengujian ketiga pada besar *lift ratio* yang dihasilkan. *Lift ratio* terbesar dengan nilai 10,753 didapatkan ketika minimum support 10%. *Lift ratio* pada minimum support 10% lebih besar dari *lift ratio* dengan minimum support 15% dan juga lebih besar dari *lift ratio* dengan minimum support 20%.



Gambar 5.7 Grafik Minimum Support Lift Ratio- Diluar Jam Kerja.

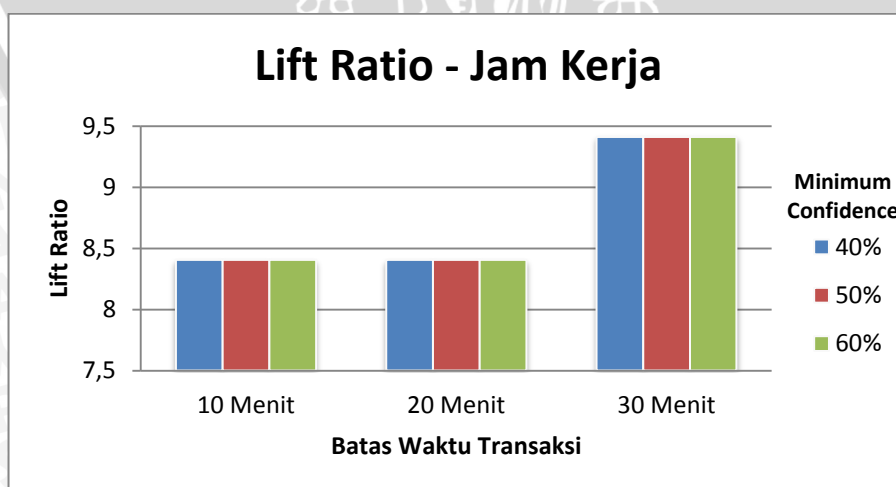
5.1.2 Hasil Pengujian Berdasarkan Minimum Confidence

Uji coba pada pengujian berdasarkan minimum *confidence* menggunakan nilai minimum *support* yang tetap yaitu sebesar 10%. Sedangkan nilai minimum *confidence* diambil secara *random*. Uji coba pertama dilakukan pada Fakultas Teknik dengan kategori waktu pada saat jam kerja.



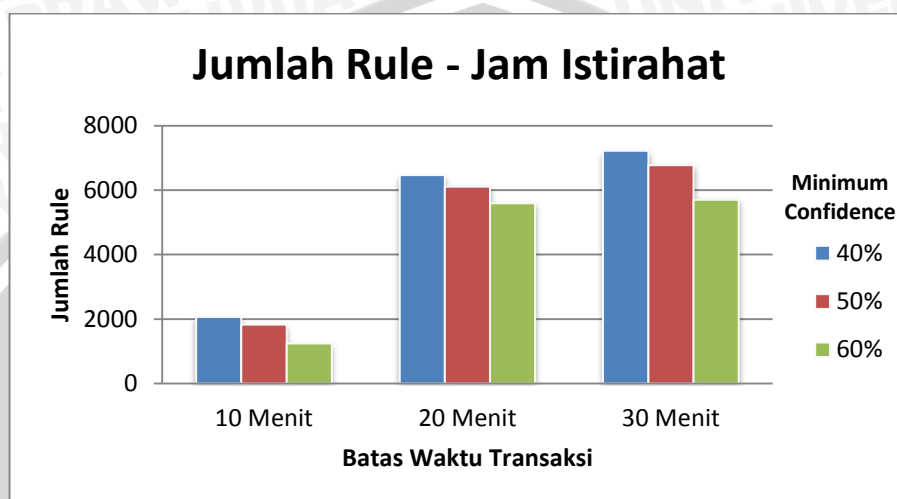
Gambar 5.8 Grafik Minimum Confidence Jumlah Rule –Jam Kerja.

Gambar 5.8 merupakan hasil pengujian pertama pada jumlah *rule* yang dihasilkan. Jumlah *rule* terbanyak sebesar 142 *rule* dihasilkan ketika nilai minimum *confidence* sebesar 40% dengan batas waktu transaksi 30 menit. Jumlah *rule* pada minimum *confidence* 40% lebih banyak dari jumlah *rule* dengan minimum *confidence* 50% dan juga lebih banyak dari jumlah *rule* dengan minimum *confidence* 60%.



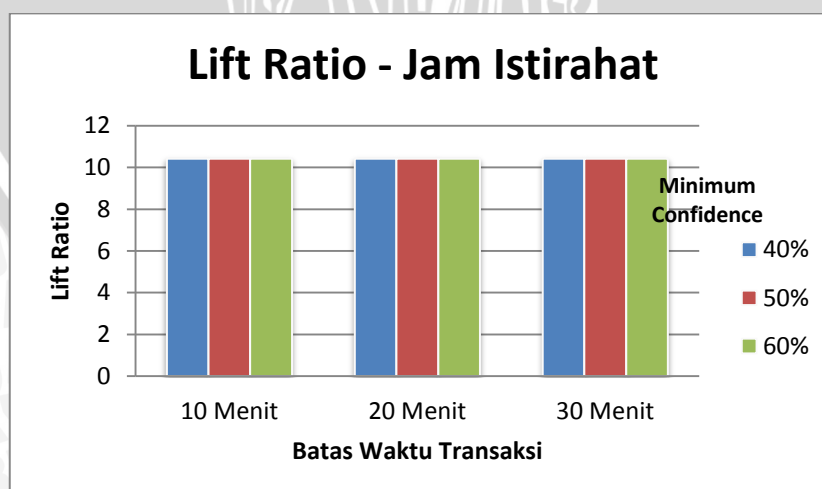
Gambar 5.9 Grafik Minimum Confidence Lift Ratio –Jam Kerja.

Gambar 5.9 merupakan hasil pengujian pertama pada besar *lift ratio* yang dihasilkan. *Lift ratio* terbesar dengan nilai 9,409 didapatkan ketika batas waktu transaksi 30 menit. *Lift ratio* pada minimum *confidence* 40% , 50% dan 60% memiliki besar yang sama.



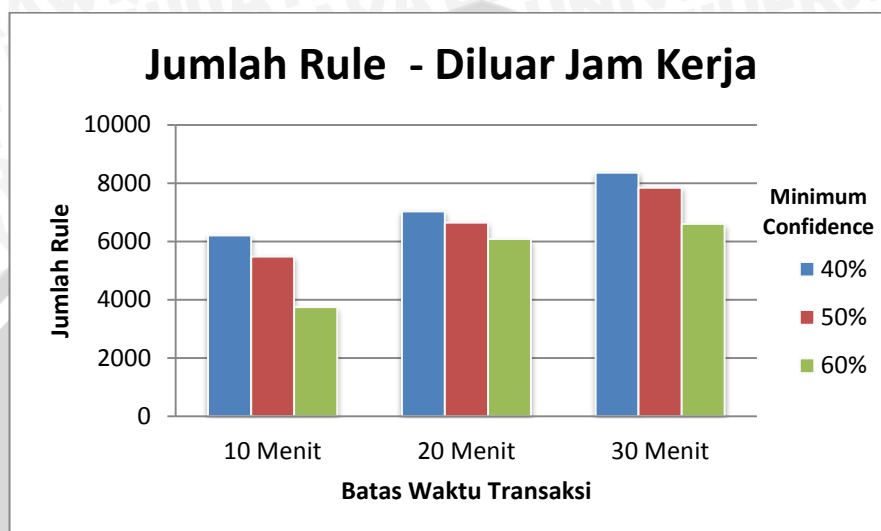
Gambar 5.10 Grafik Minimum *Confidence* Jumlah *Rule* –Jam Istirahat.

Gambar 5.10 merupakan hasil pengujian kedua pada jumlah *rule* yang dihasilkan. Jumlah *rule* terbanyak sebesar 7223 *rule* dihasilkan ketika nilai minimum *confidence* sebesar 40% dengan batas waktu transaksi 30 menit. Jumlah *rule* pada minimum *confidence* 40% lebih banyak dari jumlah *rule* dengan minimum *confidence* 50% dan juga lebih banyak dari jumlah *rule* dengan minimum *confidence* 60%.



Gambar 5.11 Grafik Minimum *Confidence* *Lift Ratio* –Jam Istirahat.

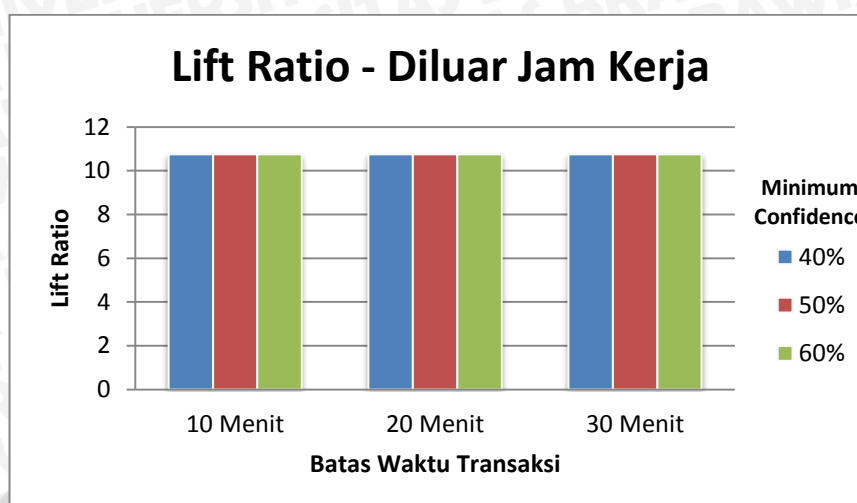
Gambar 5.11 merupakan hasil pengujian kedua pada besar *lift ratio* yang dihasilkan. *Lift ratio* pada percobaan kali ini memiliki besar yang sama yaitu 10,417.



Gambar 5.12 Grafik Minimum Confidence Jumlah Rule – Diluar Jam Kerja.

Gambar 5.12 merupakan hasil pengujian ketiga pada jumlah *rule* yang dihasilkan. Jumlah *rule* terbanyak sebesar 8362 *rule* dihasilkan ketika nilai minimum confidence sebesar 40% dengan batas waktu transaksi 30 menit. Jumlah *rule* pada minimum confidence 40% lebih banyak dari jumlah *rule* dengan minimum confidence 50% dan juga lebih banyak dari jumlah *rule* dengan minimum confidence 60%.

Gambar 5.13 merupakan hasil pengujian ketiga pada besar *lift ratio* yang dihasilkan. *Lift ratio* pada percobaan kali ini memiliki besar yang sama yaitu 10,753.



Gambar 5.13 Grafik Minimum Confidence Lift Ratio –Diluar Jam Kerja.

5.1.3 Hasil Pengujian Akurasi Algoritma SPADE

Pengujian akurasi dilakukan dengan memprosentasekan jumlah *rule* yang kuat yaitu *rule* yang memiliki nilai lift ratio lebih dari 1 terhadap jumlah *rule* keseluruhan yang terbentuk. *Rule* yang digunakan untuk pengujian akurasi algoritma SPADE adalah *rule* yang dihasilkan dari pengujian berdasarkan minimum *support* dan minimum *confidence*. Dari 54 percobaan yang telah dilakukan didapatkan rata-rata akurasi sebesar 85%. Hasil pengujian akurasi algoritma SPADE dilampirkan pada lampiran B.

5.1.4 Hasil Pengujian Rule Terbaik

Pengujian *rule* terbaik dilakukan di tiap-tiap fakultas dengan menggunakan nilai minimum *confidence* sebesar 50 % dan minimum *support* terbesar dari tiap-tiap fakultas serta batas waktu transaksi yang digunakan adalah 10 menit. Hasil pengujian untuk mendapatkan *rule* terbaik secara keseluruhan dilampirkan pada lampiran C.

5.1.4.1 Fakultas Teknik

Tabel 5.1 merupakan *rule* dengan *lift ratio* tertinggi dari setiap kategori waktu pada pengujian *rule* terbaik Fakultas Teknik. *Lift ratio* tertinggi pada

Fakultas Teknik terdapat pada kategori waktu jam kerja dengan *rule* jika mengakses blogspot.com maka mengakses blogger.com atau sebaliknya.

Tabel 5.1 Tabel *rule* terbaik Fakultas Teknik.

KATEGORI WAKTU	RULES	CONFIDENCE	LIFT RATIO
Jam Kerja	blogspot.com , blogger.com	0.917	5.804
Jam Kerja	blogger.com , blogspot.com	0.917	5.804
Jam Istirahat	yahoo.id.com , google.com	0.889	1.037
Diluar Jam Kerja	youtube.com , gstatic.com	0.767	1.657
Diluar Jam Kerja	gstatic.com , youtube.com	0.921	1.657

Sumber: Pengujian.

5.1.4.2 Fakultas Matematika dan Ilmu Pengetahuan Alam

Tabel 5.2 merupakan *rule* dengan *lift ratio* tertinggi dari setiap kategori waktu pada pengujian *rule* terbaik Fakultas Matematika dan Ilmu Pengetahuan Alam. *Lift ratio* tertinggi pada Fakultas Matematika dan Ilmu Pengetahuan Alam terdapat pada kategori waktu jam kerja dengan *rule* jika mengakses facebook.com maka mengakses google.com.

Tabel 5.2 Tabel *rule* terbaik Fakultas MIPA.

KATEGORI WAKTU	RULES	CONFIDENCE	LIFT RATIO
Jam Kerja	facebook.com , google.com	0.6	6.742
Jam Istirahat	google.com , fbcdn.net , facebook.com -> fbcdn.net	0.91	2.209
Diluar Jam Kerja	mozilla.com , bbc.co.uk	0.715	2.709

Sumber: Pengujian.

5.1.4.3 Fakultas Teknologi Pertanian

Tabel 5.3 merupakan *rule* dengan *lift ratio* tertinggi dari setiap kategori waktu pada pengujian *rule* terbaik Fakultas Teknologi Pertanian. Pada tabel tersebut hanya ada kategori waktu istirahat dan diluar jam kerja dikarenakan pada kategori waktu jam kerja tidak terbentuk *rule*. Hal ini dikarenakan pada jam kerja pengguna *internet* mengakses suatu halaman *website* yang berbeda-beda, sehingga tidak terbentuk pola pada kategori waktu tersebut.

Tabel 5.3 Tabel *rule* terbaik Fakultas Teknologi Pertanian.

KATEGORI WAKTU	RULES	CONFIDENCE	LIFT RATIO
Jam Istirahat	facebook.com , google.com , fbcnd.net	0.858	2.909
Diluar Jam Kerja	youtube.com , gstatic.com -> google.com	1.0	1.249
Diluar Jam Kerja	twitter.com -> google.com	1.0	1.249
Diluar Jam Kerja	youtube.com -> google.com	1.0	1.249
Diluar Jam Kerja	google.com -> blogspot.com	1.0	1.249
Diluar Jam Kerja	blogger.com , blogspot.com	1.0	1.249

Sumber: Pengujian.

5.1.4.4 Fakultas Kedokteran

Tabel 5.4 merupakan *rule* dengan *lift ratio* tertinggi dari setiap kategori waktu pada pengujian *rule* terbaik Fakultas Kedokteran. Pada tabel tersebut hanya ada kategori waktu jam kerja dikarenakan pada kategori waktu istirahat dan jam diluar kerja tidak terbentuk *rule*. Hal ini dikarenakan sedikitnya atau bahkan tidak ada penggunaan akses *internet* pada jam istirahat dan diluar jam kerja oleh pengguna *internet*.

Tabel 5.4 Tabel *rule* terbaik Fakultas Kedokteran.

KATEGORI WAKTU	RULES	CONFIDENCE	LIFT RATIO
Jam Kerja	facemoods.com -> entrust.net	1.0	4.0

Sumber: Pengujian.

5.1.4.5 Fakultas Pertanian

Tabel 5.5 merupakan *rule* dengan *lift ratio* tertinggi dari setiap kategori waktu pada pengujian *rule* terbaik Fakultas Pertanian. *Lift ratio* tertinggi pada Fakultas Pertanian terdapat pada kategori waktu jam kerja dengan *rule* jika mengakses twitter.com maka mengakses yahoo.id.com.

Tabel 5.5 Tabel *rule* terbaik Fakultas Pertanian.

KATEGORI WAKTU	RULES	CONFIDENCE	LIFT RATIO
Jam Kerja	twitter.com , yahoo.id.com	0.5	7.813
Jam Istirahat	bbc.co.uk , mozilla.com	1.0	2.47
Diluar Jam Kerja	facebook.com , google.com	0.734	1.53

Sumber: Pengujian.

5.1.4.6 Fakultas Peternakan

Tabel 5.6 merupakan *rule* dengan *lift ratio* tertinggi dari setiap kategori waktu pada pengujian *rule* terbaik Fakultas Peternakan. *Lift ratio* tertinggi pada Fakultas Peternakan terdapat pada kategori waktu jam kerja dengan *rule* jika mengakses mozilla.com maka mengakses bbc.co.uk atau sebaliknya.

Tabel 5.6 Tabel *rule* terbaik Fakultas Peternakan.

KATEGORI WAKTU	RULES	CONFIDENCE	LIFT RATIO
Jam Kerja	mozilla.com , bbc.co.uk	1.0	18.868
Jam Kerja	bbc.co.uk , mozilla.com	1.0	18.868
Jam Istirahat	facebook.com , google.com	0.875	1.75
Diluar Jam Kerja	google.com , facebook.com	0.91	1.024

Sumber: Pengujian.

5.1.4.7 Fakultas Perikanan dan Ilmu Kelautan

Tabel 5.7 merupakan *rule* dengan *lift ratio* tertinggi dari setiap kategori waktu pada pengujian *rule* terbaik Fakultas Perikanan dan Ilmu Kelautan. Pada tabel tersebut hanya ada kategori waktu jam kerja dan jam istirahat dikarenakan pada kategori waktu diluar jam kerja tidak terbentuk *rule*. Hal ini dikarenakan pada waktu diluar jam kerja pengguna *internet* mengakses suatu halaman *website* yang berbeda-beda, sehingga tidak terbentuk pola pada kategori waktu tersebut.

Tabel 5.7 Tabel *rule* terbaik Fakultas Perikanan dan Ilmu Kelautan.

KATEGORI WAKTU	RULES	CONFIDENCE	LIFT RATIO
Jam Kerja	zynga.com , facesmooch.com	1.0	5.989
Jam Kerja	zynga.com , conduitapps.com	1.0	5.989
Jam Kerja	unlock-this.com , mediafire.com	1.0	5.989
Jam Kerja	sweetim.sc.com , mediafire.com	1.0	5.989
Jam Kerja	sweetim.sc.com -> mediafire.com	1.0	5.989
Jam Kerja	mediafire.com , unlock-this.com	1.0	5.989
Jam Kerja	mediafire.com , sweetim.sc.com	1.0	5.989
Jam Kerja	facesmooch.com , zynga.com	1.0	5.989
Jam Kerja	facesmooch.com , conduitapps.com	1.0	5.989
Jam Kerja	conduitapps.com , zynga.com	1.0	5.989
Jam Kerja	conduitapps.com , facesmooch.com	1.0	5.989
Jam Istirahat	google.com , facebook.com -> facebook.com	1.0	1.799

Sumber: Pengujian.

5.2 Analisa

Dari hasil pengujian berdasarkan nilai minimum *support*, didapatkan bahwa semakin besar nilai minimum *support* maka semakin sedikit jumlah *rule* yang dihasilkan. Hal ini terjadi karena batasan *frequent itemset* semakin besar sehingga tidak semua *itemset* dapat dijadikan *rule* dan juga dikombinasikan pada *sequence* berikutnya. Selain itu, semakin kecil nilai minimum *support* maka nilai terbesar *lift ratio* akan semakin besar. Maka dari itu dapat disimpulkan bahwa nilai minimum *support* berpengaruh terhadap kekuatan *rule* yang dihasilkan. Batasan waktu transaksi juga berpengaruh terhadap jumlah *rule* yang dihasilkan. Semakin kecil atau sebentar waktu yang ditetapkan maka jumlah *rule* yang dibentuk semakin sedikit. Hal ini dikarenakan jangka waktu akses suatu halaman *website* yang lama atau tidak secepat dari batas waktu transaksi yang ditentukan.

Dari hasil pengujian berdasarkan nilai minimum *confidence* didapatkan bahwa semakin besar nilai minimum *confidence* maka semakin sedikit jumlah *rule* yang dihasilkan. Namun, nilai *lift ratio* cenderung konstan atau tidak berubah. Sehingga dapat disimpulkan bahwa nilai minimum *confidence*

berpengaruh terhadap jumlah *rule* yang dihasilkan namun tidak berpengaruh terhadap kekuatan *rule* yang dihasilkan.

Pada kategori waktu jam kerja, jangka waktu akses yang lama oleh pengguna *internet* terhadap suatu halaman *website* menyebabkan jumlah *rule* yang dihasilkan lebih sedikit dibandingkan jumlah *rule* pada kategori waktu jam istirahat dan diluar jam kerja. Selain itu banyaknya jumlah *rule* pada kategori waktu jam istirahat dan diluar jam kerja karena pada waktu tersebut blokir terhadap beberapa halaman *website* dibuka sehingga variasi halaman *website* yang diakses semakin banyak dibandingkan saat jam kerja.

Pengujian berdasarkan minimum *support* dan *confidence* juga menghasilkan akurasi algoritma SPADE terhadap pengolahan data sekuensial *access logs*. Dari 54 kali percobaan, dihasilkan rata-rata akurasi sebesar 85% akurat bahwa algoritma SPADE dapat digunakan untuk mencari pola perilaku pengguna *internet* menggunakan data sekuensial *access logs*.

Dari pengujian *rule* terbaik dihasilkan kekuatan *rule* terbesar dengan nilai 18,868 pada kategori waktu jam kerja di Fakultas Peternakan dengan *rule* jika mengakses mozilla.com maka mengakses bbc.co.uk dan begitu juga sebaliknya. Pengujian *rule* terbaik juga membuktikan bahwa pola perilaku setiap fakultas berbeda-beda. Contoh pada jam kerja Fakultas Teknik, dihasilkan *rule* jika mengakses blogspot.com maka mengakses blogger.com dengan kekuatan *rule* sebesar 5,084 ,sedangkan pada jam kerja Fakultas Pertanian dihasilkan *rule* jika mengakses twitter.com maka mengakses yahoo.id.com dengan kekuatan *rule* sebesar 7,813. Hal tersebut dapat disimpulkan bahwa *rule-rule* yang dihasilkan setiap fakultas di setiap waktu berbeda-beda sehingga pola perilaku pengguna *internet* di setiap fakultas pun juga berbeda-beda.

Tidak terdapatnya *rule* pada kategori waktu tertentu di beberapa fakultas terjadi dikarenakan halaman *website* yang diakses oleh pengguna *internet* berbeda-beda sehingga tidak menemukan suatu pola atau kecenderungan akses suatu halaman *website*. Selain itu sedikitnya pengguna *internet* yang mengakses halaman *website* juga menjadi penyebab tidak ditemukannya suatu pola.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

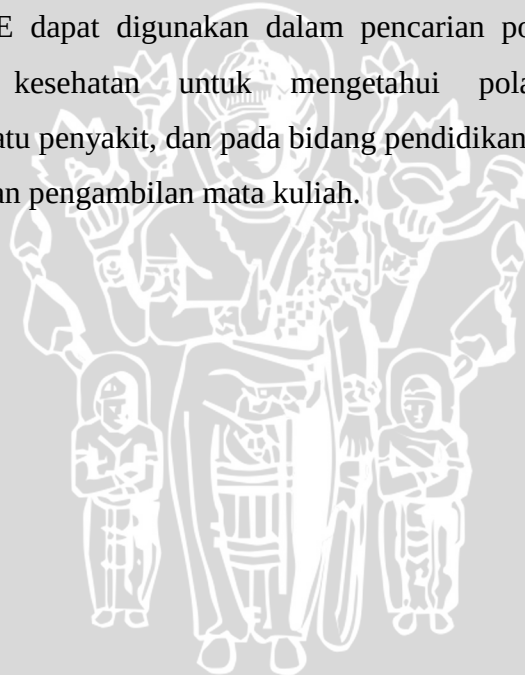
Dari hasil uji coba yang telah dilakukan maka, dapat disimpulkan sebagai berikut:

1. Algoritma SPADE (*sequential pattern discovery using equivalent classes*) diterapkan untuk mencari pola perilaku pengguna *internet* dengan cara melakukan *preprocessing* data untuk menyaring informasi yang dibutuhkan. Dilanjutkan dengan pembentukan data transaksi dan perhitungan SPADE dengan mengkombinasikan *itemset* dan menghitung *frequent*-nya untuk mendapatkan *rule* yang kemudian akan dicari kekuatan setiap *rule* dengan menghitung *lift ratio*-nya.
2. Tingkat kekuatan atau *lift ratio* dari *rule* yang dihasilkan pada tiap fakultas berbeda-beda. Nilai *lift ratio* tertinggi pada Fakultas Teknik sebesar 5,804 dengan salah satu *rule*-nya jika mengakses blogspot.com maka mengakses blogger.com. Nilai *lift ratio* tertinggi pada Fakultas MIPA sebesar 6,742 dengan *rule* jika mengakses facebook.com maka mengakses google.com. Nilai *lift ratio* tertinggi pada Fakultas Teknologi Pertanian sebesar 2,909 dengan *rule* jika mengakses facebook.com dan google.com maka mengakses fbcdn.net. Nilai *lift ratio* tertinggi pada Fakultas Kedokteran sebesar 4,0 dengan *rule* jika mengakses facemoods.com maka mengakses entrust.net. Nilai *lift ratio* tertinggi pada Fakultas Pertanian sebesar 7,813 dengan *rule* jika mengakses twitter.com maka mengakses yahoo.id.com. Nilai *lift ratio* tertinggi pada Fakultas Peternakan sebesar 18,868 dengan *rule* jika mengakses mozilla.com maka mengakses bbc.co.uk. Sedangkan nilai *lift ratio* tertinggi pada Fakultas Perikanan dan Ilmu Kelautan sebesar 5,989 dengan salah satu *rule*-nya jika mengakses zynga.com maka mengakses facesmooch.com. Perbedaan nilai *lift ratio* pada setiap fakultas dikarenakan perbedaan jumlah pengguna *internet* yang mengakses *internet* dan besar nilai *confidence* maupun nilai *support* dari *rule* itu sendiri.

6.2 Saran

Saran yang dapat diajukan setelah menyelesaikan penelitian ini adalah sebagai berikut:

1. Data yang digunakan untuk menemukan pola perilaku pengguna *internet* dapat beragam atau dalam jumlah besar untuk menemukan pola yang lebih beragam.
2. Pada saat *preprocessing* data, sistem perlu melakukan *filtering* terhadap halaman-halaman *website* iklan atau *advertise* dengan cara *parsing domain* *advertise* atau *tracking* alamat *website* tersebut menggunakan *information retrieval* (IR). Dikarenakan halaman-halaman *website* iklan tidak dibutuhkan pada pencarian pola perilaku pengguna *internet*.
3. Algoritma SPADE dapat digunakan dalam pencarian pola di bidang lain seperti bidang kesehatan untuk mengetahui pola kecenderungan pendiagnosaan suatu penyakit, dan pada bidang pendidikan untuk mengetahui pola kecenderungan pengambilan mata kuliah.



DAFTAR PUSTAKA

- [AGR-93] Agrawal, R., Srikant, R. 1993. *Mining Sequential Patterns*. San Jose: IBM Almaden Research Center.
- [AMP-03] Amperiyanto, Tri. 2003. *Bermain-main dengan Internet*. Jakarta: PT Elex Media Komputindo.
- [ARD-13] Ardiansyah, R. 2013. *Sequential Pattern Mining pada Data Transaksi Penjualan Menggunakan Algoritma Sequential Pattern Discovery Using Equivalent Classes (SPADE)*. Skripsi tidak diterbitkan. Malang : Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
- [ART-03] Ardiantoro, Dinding. 2003. Pengantar DNS (Domain Name System). (Online) (<http://ptik.unhas.ac.id/tahir/BAHAN-KULIAH/TEK.%20JARINGAN%20KOMPUTER%20-%20TE/jaringan-dan-sekuriti/diding-dns.pdf> diakses tanggal 1 April 2014)
- [BUD-09] Budhi, G.S., Handojo, A., Wirawan, C.O. 2009. *Algoritma Generalized Sequential Pattern untuk Menggali Data Sekuensial Sirkulasi Buku Pada Perpustakaan UK Petra*. Makalah disajikan pada Seminar Nasional Aplikasi Teknologi Informasi, Yogyakarta 20 Juni.
- [HAN-01] Han, J., Kamber, M. 2001. *Data Mining: Concepts and Techniques*. California: Morgan Kauffman Publishers.
- [HID-10] Hidayat, Rahmat. 2010. *Cara Praktis Membangun Website Gratis*. Jakarta: PT Elex Media Komputindo.
- [KAN-03] Kantardzic, Mehmed. 2003. *Data Mining : Concepts, Models, Methods, and Algorithms*. New Jersey: John Willey & Sons Inc.
- [KUS-07] Kusrini. 2007. *Konsep dan Aplikasi Sistem Pendukung Keputusan*. Yogyakarta: Andi.
- [LEU-07] Leung, Ming Kok. 2007. *Association Rules*. New York: Polytechnic Institute of New York.
- [MAU-07] Maulana, Heri D.J. 2007. *Promosi Kesehatan*. Jakarta: Buku Kedokteran EGC.
- [MCL-07] McLeod, R., Schell, G.P. 2007. *Sistem Informasi Manajemen Edisi 10*. Jakarta: Salemba Empat.

- [NOV-13] Novianto, Iik. 2013. Perilaku Penggunaan Internet di Kalangan Mahasiswa (Studi deskriptif tentang perilaku penggunaan internet dikalangan mahasiswa perguruan tinggi negeri (FISIP UNAIR) dengan perguruan tinggi swasta (FISIP UPN) untuk memenuhi kebutuhan informasinya). *Jurnal Universitas Airlangga*. (Online) (<http://journal.unair.ac.id/filerPDF/Jurnal%20IIK%20Novianto.pdf> diakses tanggal 29 Januari 2014)
- [PEI-01] Pei, J., dkk. 2001. *PrefixSpan: Mining Sequential Patterns Efficiently by Prefix Projected Pattern Growth*. Natural Science and Engineering Research Council of Canada (grant NSERC-A3723), the Networks of Centres of Excellence of Canada (grant NCE/IRIS-3), and Hawlett-Packard Lab, USA.
- [PER-08] Perdan, Fredy. 2008. *Sistem Rekomendasi untuk Pengaturan Hyperlink pada Halaman Utama Website menggunakan Metode Association Rule dan Hyperlink Clustering*. Penelitian tidak diterbitkan. Malang: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Brawijaya.
- [PET-04] Peter, Hofegesang. 2004. *Web Usage Mining Structuring Semantically Enriched Click Stream Data*. Penelitian tidak diterbitkan. Amsterdam: Universitas Vrije.
- [PUB-13] Website Official Universitas Brawijaya. 2013. *Mahasiswa Universitas Brawijaya*. (Online) (<http://ub.ac.id/tentang/profil-universitas/mahasiswa> diakses tanggal 27 Januari 2014).
- [RAH-03] Rahayu, Lisda. 2003. *Kajian mengenai Perilaku Peneliti dalam Memanfaatkan dan Menyetir Sumber Informasi Berbasis Informasi Berbasis Internet* (proposal penelitian). Thesis tidak diterbitkan. Jakarta : Program Pasca Sarjana Fakultas Ilmu Budaya Universitas Indonesia.
- [RAM-05] Ramadhan, Arief. 2005. *Eri Pelajaran Komputer: Internet dan Aplikasinya*. Jakarta: PT Elex Media Komputindo.
- [RAS-08] Rasul, Juharis. 2008. *Teknologi Informasi dan Komunikasi 2 SMA Kelas XI*. Surabaya: Quadra Inti Solusi.
- [RUL-08] Ruldeviyani, Y., Fahrian, M. 2008. *Implementasi Algoritma-algoritma Association Rules sebagai Bagian dari pengembangan Data Mining Algorithms Collection*. Jakarta: Fakultas Ilmu Komputer Universitas Indonesia.
- [SAN-07] Santosa, Budi. 2007. *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta: Graha Ilmu.

- [SIG-06] Sigit, Kusworo. 2006. *Pemodelan Tingkah Laku Pengunjung Situs Web Berdasarkan Data Log Web Server*.
- [SOE-06] Soeroso, Andreas. 2006. *Sosiologi I*. Jakarta:Yudhistira.
- [SOL-09] Sholichah, Alfiyatus. 2009. *Data Mining untuk Pembiayaan Murabahah menggunakan Association Rule*. Skripsi tidak diterbitkan. Malang : Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim.
- [SUN-07] Sunarto. 2007. *Teknologi Informasi dan Komunikasi IX*. Jakarta: Grasindo.
- [SUY-10] Suyanto, Mohammad. 2010. Universal Resource Locataor (URL). *Karya Ilmiah Dosen Amikom*. (Online) (<http://www.amikom.ac.id/research/index.php/karyailmiahdosen/article/view/2122> diakses tanggal 2 April 2014)
- [SYA-05] Syafrizal, Melwin. 2005. *Pengantar Jaringan Komputer*. Yogyakarta: Andi.
- [WIJ-13] Wijaya, Mila Febri. 2013. *Pencarian Pola Perilaku Pengguna Internet Menggunakan Metode Association Rule Mining, Studi Kasus FMIPA Universitas Brawijaya*. Skripsi tidak diterbitkan. Malang : Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
- [YUH-05] Yuhefizar, dkk. 2005. *Cara Mudah Membangun Website Interaktif Menggunakan Content Management System Joomla*. Jakarta: PT Elex Media Komputindo.

LAMPIRAN A

Uji coba berdasarkan minimum *support* dan minimum *confidence*.

No	Kategori Waktu	Batas Waktu (Menit)	Minimal Support	Minimal Confidence	Lift Ratio
1	Jam Kerja	10	10%	50%	8,404
2	Jam Kerja	10	15%	50%	5,804
3	Jam Kerja	10	20%	50%	0
4	Jam Kerja	20	10%	50%	8,404
5	Jam Kerja	20	15%	50%	5,804
6	Jam Kerja	20	20%	50%	0
7	Jam Kerja	30	10%	50%	9,409
8	Jam Kerja	30	15%	50%	5,804
9	Jam Kerja	30	20%	50%	0
10	Jam Istirahat	10	10%	50%	10,417
11	Jam Istirahat	10	15%	50%	6,994
12	Jam Istirahat	10	20%	50%	3,226
13	Jam Istirahat	20	10%	50%	10,417
14	Jam Istirahat	20	15%	50%	6,994
15	Jam Istirahat	20	20%	50%	3,352
16	Jam Istirahat	30	10%	50%	10,417
17	Jam Istirahat	30	15%	50%	6,994
18	Jam Istirahat	30	20%	50%	4,194
19	Diluar Jam Kerja	10	10%	50%	10,753
20	Diluar Jam Kerja	10	15%	50%	4,485
21	Diluar Jam Kerja	10	20%	50%	4,144
22	Diluar Jam Kerja	20	10%	50%	10,753
23	Diluar Jam Kerja	20	15%	50%	4,485
24	Diluar Jam Kerja	20	20%	50%	3,74
25	Diluar Jam Kerja	30	10%	50%	10,753
26	Diluar Jam Kerja	30	15%	50%	4,485
27	Diluar Jam Kerja	30	20%	50%	3,74
28	Jam Kerja	10	10%	40%	8,404
29	Jam Kerja	10	10%	50%	8,404
30	Jam Kerja	10	10%	60%	8,404
31	Jam Kerja	20	10%	40%	8,404
32	Jam Kerja	20	10%	50%	8,404
33	Jam Kerja	20	10%	60%	8,404
34	Jam Kerja	30	10%	40%	9,409

35	Jam Kerja	30	10%	50%	9,409
36	Jam Kerja	30	10%	60%	9,409
37	Jam Istirahat	10	10%	40%	10,417
38	Jam Istirahat	10	10%	50%	10,417
39	Jam Istirahat	10	10%	60%	10,417
40	Jam Istirahat	20	10%	40%	10,417
41	Jam Istirahat	20	10%	50%	10,417
42	Jam Istirahat	20	10%	60%	10,417
43	Jam Istirahat	30	10%	40%	10,417
44	Jam Istirahat	30	10%	50%	10,417
45	Jam Istirahat	30	10%	60%	10,417
46	Diluar Jam Kerja	10	10%	40%	10,753
47	Diluar Jam Kerja	10	10%	50%	10,753
48	Diluar Jam Kerja	10	10%	60%	10,753
49	Diluar Jam Kerja	20	10%	40%	10,753
50	Diluar Jam Kerja	20	10%	50%	10,753
51	Diluar Jam Kerja	20	10%	60%	10,753
52	Diluar Jam Kerja	30	10%	40%	10,753
53	Diluar Jam Kerja	30	10%	50%	10,753
54	Diluar Jam Kerja	30	10%	60%	10,753

LAMPIRAN B

Uji coba akurasi algoritma SPADE.

No	Kategori Waktu	Batas Waktu (Menit)	Minimal Support	Minimal Confidence	Jumlah Rule (x)	Jumlah Rule Kuat (y)	Akurasi (y/x)*100%
1	Jam Kerja	10	10%	50%	98	98	100%
2	Jam Kerja	10	15%	50%	13	13	100%
3	Jam Kerja	10	20%	50%	0	0	0%
4	Jam Kerja	20	10%	50%	86	86	100%
5	Jam Kerja	20	15%	50%	9	9	100%
6	Jam Kerja	20	20%	50%	0	0	0%
7	Jam Kerja	30	10%	50%	132	132	100%
8	Jam Kerja	30	15%	50%	12	12	100%
9	Jam Kerja	30	20%	50%	0	0	0%
10	Jam Istirahat	10	10%	50%	1829	1616	88%
11	Jam Istirahat	10	15%	50%	354	284	80%
12	Jam Istirahat	10	20%	50%	75	50	67%
13	Jam Istirahat	20	10%	50%	6108	5516	90%
14	Jam Istirahat	20	15%	50%	703	601	85%
15	Jam Istirahat	20	20%	50%	148	118	80%
16	Jam Istirahat	30	10%	50%	6779	6308	93%
17	Jam Istirahat	30	15%	50%	880	801	91%
18	Jam Istirahat	30	20%	50%	171	155	91%
19	Diluar Jam Kerja	10	10%	50%	5480	4873	89%
20	Diluar Jam Kerja	10	15%	50%	1115	884	79%
21	Diluar Jam Kerja	10	20%	50%	565	427	76%
22	Diluar Jam Kerja	20	10%	50%	6664	5622	84%
23	Diluar Jam Kerja	20	15%	50%	1296	1031	80%
24	Diluar Jam Kerja	20	20%	50%	611	460	75%
25	Diluar Jam Kerja	30	10%	50%	7848	6371	81%
26	Diluar Jam Kerja	30	15%	50%	1713	1357	79%
27	Diluar Jam Kerja	30	20%	50%	866	653	75%
28	Jam Kerja	10	10%	40%	111	111	100%

29	Jam Kerja	10	10%	50%	98	98	100%
30	Jam Kerja	10	10%	60%	67	67	100%
31	Jam Kerja	20	10%	40%	103	103	100%
32	Jam Kerja	20	10%	50%	86	86	100%
33	Jam Kerja	20	10%	60%	69	69	100%
34	Jam Kerja	30	10%	40%	142	142	100%
35	Jam Kerja	30	10%	50%	132	132	100%
36	Jam Kerja	30	10%	60%	107	107	100%
37	Jam Istirahat	10	10%	40%	2071	1787	86%
38	Jam Istirahat	10	10%	50%	1829	1616	88%
39	Jam Istirahat	10	10%	60%	1248	1132	91%
40	Jam Istirahat	20	10%	40%	6464	5779	89%
41	Jam Istirahat	20	10%	50%	6108	5516	90%
42	Jam Istirahat	20	10%	60%	5594	5102	91%
43	Jam Istirahat	30	10%	40%	7223	6655	92%
44	Jam Istirahat	30	10%	50%	6779	6308	93%
45	Jam Istirahat	30	10%	60%	5700	5358	94%
46	Diluar Jam Kerja	10	10%	40%	6205	5336	86%
47	Diluar Jam Kerja	10	10%	50%	5480	4873	89%
48	Diluar Jam Kerja	10	10%	60%	3739	3365	90%
49	Diluar Jam Kerja	20	10%	40%	7031	6257	89%
50	Diluar Jam Kerja	20	10%	50%	6644	5622	85%
51	Diluar Jam Kerja	20	10%	60%	6084	5536	91%
52	Diluar Jam Kerja	30	10%	40%	8362	7693	92%
53	Diluar Jam Kerja	30	10%	50%	7848	6371	81%
54	Diluar Jam Kerja	30	10%	60%	6598	6202	94%
RATA-RATA							85%

LAMPIRAN C

Uji coba rule terbaik.

NO	FAKULTAS	KATEGORI WAKTU	RULES	CONFIDENCE	LIFT RATIO
1	FPIK	Jam Kerja	zynga.com , facesmooch.com	1.0	5.989
2	FPIK	Jam Kerja	zynga.com , conduitapps.com	1.0	5.989
3	FPIK	Jam Kerja	yieldmanager.ad.com -> gamepaylabs.com	1.0	4.0
4	FPIK	Jam Kerja	unlock-this.com , mediafire.com	1.0	5.989
5	FPIK	Jam Kerja	unlock-this.com , gamepaylabs.com	1.0	4.0
6	FPIK	Jam Kerja	sweetim.sc.com , mediafire.com	1.0	5.989
7	FPIK	Jam Kerja	sweetim.sc.com -> mediafire.com	1.0	5.989
8	FPIK	Jam Kerja	sweetim.sc.com -> gamepaylabs.com	1.0	4.0
9	FPIK	Jam Kerja	mediafire.com , unlock-this.com	1.0	5.989
10	FPIK	Jam Kerja	mediafire.com , sweetim.sc.com	1.0	5.989
11	FPIK	Jam Kerja	mediafire.com , gamepaylabs.com	1.0	4.0
12	FPIK	Jam Kerja	mediafire.com -> gamepaylabs.com	1.0	4.0
13	FPIK	Jam Kerja	gamepaylabs.com , unlock-this.com	0.667	3.995
14	FPIK	Jam Kerja	gamepaylabs.com , mediafire.com	0.667	3.995
15	FPIK	Jam Kerja	facesmooch.com , zynga.com	1.0	5.989
16	FPIK	Jam Kerja	facesmooch.com , conduitapps.com	1.0	5.989
17	FPIK	Jam Kerja	conduitapps.com , zynga.com	1.0	5.989
18	FPIK	Jam Kerja	conduitapps.com , facesmooch.com	1.0	5.989
19	FPIK	Jam Istirahat	google.com , facebook.com -> facebook.com	1.0	1.799
20	FPIK	Jam Istirahat	google.com , facebook.com	0.801	1.441
21	FPIK	Jam Istirahat	google.com -> facebook.com	0.801	1.441
22	FPIK	Jam Istirahat	facebook.com , google.com	0.801	1.441
23	Peternakan	Jam Kerja	mozilla.com , bbc.co.uk	1.0	18.868

24	Peternakan	Jam Kerja	bbc.co.uk , mozilla.com	1.0	18.868
25	Peternakan	Jam Istirahat	google.com , facebook.com	1.0	1.749
26	Peternakan	Jam Istirahat	google.com -> facebook.com	0.858	1.5
27	Peternakan	Jam Istirahat	facebook.com , google.com	0.875	1.75
28	Peternakan	Diluar Jam Kerja	mozilla.com , facebook.com	0.847	0.953
29	Peternakan	Diluar Jam Kerja	mozilla.com -> facebook.com	0.924	1.04
30	Peternakan	Diluar Jam Kerja	google.com , facebook.com	0.91	1.024
31	Peternakan	Diluar Jam Kerja	facebook.com , mozilla.com	0.688	0.952
32	Peternakan	Diluar Jam Kerja	facebook.com , google.com	0.625	1.022
33	Pertanian	Jam Kerja	twitter.com , yahoo.id.com	0.5	7.813
34	Pertanian	Jam Istirahat	mozilla.com , google.com	0.948	1.143
35	Pertanian	Jam Istirahat	mozilla.com , bbc.co.uk	0.79	2.469
36	Pertanian	Jam Istirahat	google.com , facebook.com	0.744	0.972
37	Pertanian	Jam Istirahat	google.com -> facebook.com	0.616	0.805
38	Pertanian	Jam Istirahat	facebook.com , google.com	0.806	0.972
39	Pertanian	Jam Istirahat	bbc.co.uk , mozilla.com	1.0	2.47
40	Pertanian	Diluar Jam Kerja	google.com , facebook.com	0.917	1.529
41	Pertanian	Diluar Jam Kerja	facebook.com , google.com	0.734	1.53
42	FTP	Jam Istirahat	google.com , facebook.com -> facebook.com	0.858	0.911
43	FTP	Jam Istirahat	facebook.com , google.com , fbcnd.net	0.858	2.909
44	FTP	Jam Istirahat	yahoo.id.com , facebook.com	1.0	1.062
45	FTP	Jam Istirahat	mozilla.com , google.com	0.556	0.63
46	FTP	Jam Istirahat	mozilla.com -> google.com	0.556	0.63
47	FTP	Jam Istirahat	mozilla.com , facebook.com	0.778	0.826
48	FTP	Jam Istirahat	mozilla.com -> facebook.com	0.556	0.591
49	FTP	Jam Istirahat	mozilla.com , bbc.co.uk	0.556	1.885
50	FTP	Jam Istirahat	gstatic.com , google.com	0.834	0.945
51	FTP	Jam Istirahat	googlesyndication.com , google.com	1.0	1.133
52	FTP	Jam Istirahat	google.com , facebook.com	0.934	0.992
53	FTP	Jam Istirahat	google.com -> facebook.com	0.667	0.709
54	FTP	Jam Istirahat	fbcnd.net , facebook.com	1.0	1.062
55	FTP	Jam Istirahat	facebook.com , google.com	0.875	0.991
56	FTP	Jam Istirahat	bbc.co.uk , mozilla.com	1.0	1.887

57	FTP	Diluar Jam Kerja	youtube.com , gstatic.com , googleusercontent.com	1.0	1.249
58	FTP	Diluar Jam Kerja	youtube.com , gstatic.com , google.com	1.0	1.249
59	FTP	Diluar Jam Kerja	youtube.com , gstatic.com -> google.com	1.0	1.249
60	FTP	Diluar Jam Kerja	youtube.com , gstatic.com , fbcdn.net	1.0	1.249
61	FTP	Diluar Jam Kerja	youtube.com , gstatic.com -> fbcdn.net	1.0	1.249
62	FTP	Diluar Jam Kerja	youtube.com , gstatic.com , facebook.com	1.0	1.0
63	FTP	Diluar Jam Kerja	youtube.com , gstatic.com -> facebook.com	1.0	1.0
64	FTP	Diluar Jam Kerja	youtube.com , gstatic.com , blogger.com	1.0	1.249
65	FTP	Diluar Jam Kerja	gstatic.com , youtube.com , twitter.com	1.0	1.249
66	FTP	Diluar Jam Kerja	gstatic.com , youtube.com , mozilla.com	1.0	1.249
67	FTP	Diluar Jam Kerja	gstatic.com , youtube.com , google.com	1.0	1.249
68	FTP	Diluar Jam Kerja	gstatic.com , youtube.com -> google.com	1.0	1.249
69	FTP	Diluar Jam Kerja	gstatic.com , youtube.com , fbcdn.net	1.0	1.249
70	FTP	Diluar Jam Kerja	gstatic.com , youtube.com -> fbcdn.net	1.0	1.249
71	FTP	Diluar Jam Kerja	gstatic.com , youtube.com , facebook.com	1.0	1.0
72	FTP	Diluar Jam Kerja	gstatic.com , youtube.com -> facebook.com	1.0	1.0
73	FTP	Diluar Jam Kerja	blogspot.com -> google.com , fbcdn.net	1.0	1.249
74	FTP	Diluar Jam Kerja	blogspot.com -> google.com , facebook.com	1.0	1.0
75	FTP	Diluar Jam Kerja	blogspot.com -> fbcdn.net , facebook.com	1.0	1.0
76	FTP	Diluar Jam Kerja	blogger.com -> google.com , facebook.com	1.0	1.0
77	FTP	Diluar Jam Kerja	youtube.com , twitter.com	1.0	1.249
78	FTP	Diluar Jam Kerja	youtube.com , mozilla.com	1.0	1.249
79	FTP	Diluar Jam Kerja	youtube.com , gstatic.com	1.0	1.249

80	FTP	Diluar Jam Kerja	youtube.com , googleusercontent.com	1.0	1.249
81	FTP	Diluar Jam Kerja	youtube.com , google.com	1.0	1.249
82	FTP	Diluar Jam Kerja	youtube.com -> google.com	1.0	1.249
83	FTP	Diluar Jam Kerja	youtube.com , fbcdn.net	1.0	1.249
84	FTP	Diluar Jam Kerja	youtube.com -> fbcdn.net	1.0	1.249
85	FTP	Diluar Jam Kerja	youtube.com , facebook.com	1.0	1.0
86	FTP	Diluar Jam Kerja	youtube.com -> facebook.com	1.0	1.0
87	FTP	Diluar Jam Kerja	youtube.com , blogger.com	1.0	1.249
88	FTP	Diluar Jam Kerja	twitter.com , youtube.com	1.0	1.249
89	FTP	Diluar Jam Kerja	twitter.com , gstatic.com	1.0	1.249
90	FTP	Diluar Jam Kerja	twitter.com -> google.com	1.0	1.249
91	FTP	Diluar Jam Kerja	twitter.com , fbcdn.net	1.0	1.249
92	FTP	Diluar Jam Kerja	twitter.com -> fbcdn.net	1.0	1.249
93	FTP	Diluar Jam Kerja	twitter.com , facebook.com	1.0	1.0
94	FTP	Diluar Jam Kerja	twitter.com -> facebook.com	1.0	1.0
95	FTP	Diluar Jam Kerja	mozilla.com , youtube.com	1.0	1.249
96	FTP	Diluar Jam Kerja	mozilla.com , gstatic.com	1.0	1.249
97	FTP	Diluar Jam Kerja	mozilla.com -> google.com	1.0	1.249
98	FTP	Diluar Jam Kerja	mozilla.com -> fbcdn.net	1.0	1.249
99	FTP	Diluar Jam Kerja	mozilla.com , facebook.com	1.0	1.0
100	FTP	Diluar Jam Kerja	mozilla.com -> facebook.com	1.0	1.0
101	FTP	Diluar Jam Kerja	gstatic.com , youtube.com	1.0	1.249
102	FTP	Diluar Jam Kerja	gstatic.com , twitter.com	1.0	1.249

103	FTP	Diluar Jam Kerja	gstatic.com , mozilla.com	1.0	1.249
104	FTP	Diluar Jam Kerja	gstatic.com , google.com	1.0	1.249
105	FTP	Diluar Jam Kerja	gstatic.com -> google.com	1.0	1.249
106	FTP	Diluar Jam Kerja	gstatic.com , fbcdn.net	1.0	1.249
107	FTP	Diluar Jam Kerja	gstatic.com -> fbcdn.net	1.0	1.249
108	FTP	Diluar Jam Kerja	gstatic.com , facebook.com	1.0	1.0
109	FTP	Diluar Jam Kerja	gstatic.com -> facebook.com	1.0	1.0
110	FTP	Diluar Jam Kerja	googleusercontent.com , youtube.com	1.0	1.249
111	FTP	Diluar Jam Kerja	googleusercontent.com -> twitter.com	1.0	1.249
112	FTP	Diluar Jam Kerja	googleusercontent.com , google.com	1.0	1.249
113	FTP	Diluar Jam Kerja	googleusercontent.com -> google.com	1.0	1.249
114	FTP	Diluar Jam Kerja	googleusercontent.com -> fbcdn.net	1.0	1.249
115	FTP	Diluar Jam Kerja	googleusercontent.com , facebook.com	1.0	1.0
116	FTP	Diluar Jam Kerja	googleusercontent.com -> facebook.com	1.0	1.0
117	FTP	Diluar Jam Kerja	googleusercontent.com , blogspot.com	1.0	1.249
118	FTP	Diluar Jam Kerja	googleusercontent.com , blogger.com	1.0	1.249
119	FTP	Diluar Jam Kerja	google.com , youtube.com	1.0	1.249
120	FTP	Diluar Jam Kerja	google.com , gstatic.com	1.0	1.249
121	FTP	Diluar Jam Kerja	google.com , googleusercontent.com	1.0	1.249
122	FTP	Diluar Jam Kerja	google.com -> fbcdn.net	1.0	1.249
123	FTP	Diluar Jam Kerja	google.com , facebook.com	1.0	1.0
124	FTP	Diluar Jam Kerja	google.com -> facebook.com	1.0	1.0
125	FTP	Diluar Jam Kerja	google.com , blogspot.com	1.0	1.249

126	FTP	Diluar Jam Kerja	google.com -> blogspot.com	1.0	1.249
127	FTP	Diluar Jam Kerja	google.com , blogger.com	1.0	1.249
128	FTP	Diluar Jam Kerja	google.com -> blogger.com	1.0	1.249
129	FTP	Diluar Jam Kerja	fbcdn.net , youtube.com	1.0	1.249
130	FTP	Diluar Jam Kerja	fbcdn.net , twitter.com	1.0	1.249
131	FTP	Diluar Jam Kerja	fbcdn.net , gstatic.com	1.0	1.249
132	FTP	Diluar Jam Kerja	fbcdn.net -> google.com	1.0	1.249
133	FTP	Diluar Jam Kerja	fbcdn.net , facebook.com	1.0	1.0
134	FTP	Diluar Jam Kerja	fbcdn.net -> facebook.com	1.0	1.0
135	FTP	Diluar Jam Kerja	facebook.com , youtube.com	0.801	1.0
136	FTP	Diluar Jam Kerja	facebook.com , twitter.com	0.801	1.0
137	FTP	Diluar Jam Kerja	facebook.com , mozilla.com	0.801	1.0
138	FTP	Diluar Jam Kerja	facebook.com , gstatic.com	0.801	1.0
139	FTP	Diluar Jam Kerja	facebook.com , googleusercontent.com	0.801	1.0
140	FTP	Diluar Jam Kerja	facebook.com , google.com	0.801	1.0
141	FTP	Diluar Jam Kerja	facebook.com , fbcdn.net	0.801	1.0
142	FTP	Diluar Jam Kerja	facebook.com , blogspot.com	0.801	1.0
143	FTP	Diluar Jam Kerja	facebook.com -> blogspot.com	0.801	1.0
144	FTP	Diluar Jam Kerja	facebook.com , blogger.com	0.801	1.0
145	FTP	Diluar Jam Kerja	facebook.com -> blogger.com	0.801	1.0
146	FTP	Diluar Jam Kerja	blogspot.com -> twitter.com	1.0	1.249
147	FTP	Diluar Jam Kerja	blogspot.com , googleusercontent.com	1.0	1.249
148	FTP	Diluar Jam Kerja	blogspot.com , google.com	1.0	1.249

149	FTP	Diluar Jam Kerja	blogspot.com -> google.com	1.0	1.249
150	FTP	Diluar Jam Kerja	blogspot.com -> fbcdn.net	1.0	1.249
151	FTP	Diluar Jam Kerja	blogspot.com , facebook.com	1.0	1.0
152	FTP	Diluar Jam Kerja	blogspot.com -> facebook.com	1.0	1.0
153	FTP	Diluar Jam Kerja	blogspot.com , blogger.com	1.0	1.249
154	FTP	Diluar Jam Kerja	blogger.com , youtube.com	1.0	1.249
155	FTP	Diluar Jam Kerja	blogger.com -> twitter.com	1.0	1.249
156	FTP	Diluar Jam Kerja	blogger.com , googleusercontent.com	1.0	1.249
157	FTP	Diluar Jam Kerja	blogger.com , google.com	1.0	1.249
158	FTP	Diluar Jam Kerja	blogger.com -> google.com	1.0	1.249
159	FTP	Diluar Jam Kerja	blogger.com , facebook.com	1.0	1.0
160	FTP	Diluar Jam Kerja	blogger.com -> facebook.com	1.0	1.0
161	FTP	Diluar Jam Kerja	blogger.com , blogspot.com	1.0	1.249
162	FK	Jam Kerja	facemoods.com -> entrust.net	1.0	4.0
163	MIPA	Jam Kerja	facebook.com , google.com	0.6	6.742
164	MIPA	Jam Istirahat	google.com , fbcdn.net , facebook.com -> fbcdn.net	0.91	2.209
165	MIPA	Jam Istirahat	google.com , fbcdn.net , facebook.com -> facebook.com	0.91	1.145
166	MIPA	Jam Istirahat	google.com , fbcdn.net -> fbcdn.net	0.847	2.056
167	MIPA	Jam Istirahat	google.com , fbcdn.net , facebook.com	0.847	1.066
168	MIPA	Jam Istirahat	google.com , fbcdn.net -> facebook.com	0.847	1.066
169	MIPA	Jam Istirahat	google.com , facebook.com -> fbcdn.net	0.834	2.025
170	MIPA	Jam Istirahat	google.com , facebook.com -> facebook.com	0.834	1.05
171	MIPA	Jam Istirahat	fbcdn.net , google.com , facebook.com	0.847	1.066
172	MIPA	Jam Istirahat	facebook.com , google.com ,	0.834	2.025

			fbcnd.net		
173	MIPA	Jam Istirahat	twitter.com , google.com	0.834	1.182
174	MIPA	Jam Istirahat	twitter.com , facebook.com	0.917	1.154
175	MIPA	Jam Istirahat	mozilla.com , google.com	0.737	1.044
176	MIPA	Jam Istirahat	mozilla.com -> google.com	0.632	0.896
177	MIPA	Jam Istirahat	mozilla.com , facebook.com	0.685	0.862
178	MIPA	Jam Istirahat	mozilla.com -> facebook.com	0.737	0.928
179	MIPA	Jam Istirahat	googleusercontent.com , facebook.com	1.0	1.258
180	MIPA	Jam Istirahat	google.com , mozilla.com	0.584	1.045
181	MIPA	Jam Istirahat	google.com , fbcnd.net	0.542	1.316
182	MIPA	Jam Istirahat	google.com , facebook.com	0.75	0.944
183	MIPA	Jam Istirahat	google.com -> facebook.com	0.625	0.787
184	MIPA	Jam Istirahat	fbcnd.net , google.com	0.929	1.316
185	MIPA	Jam Istirahat	fbcnd.net , facebook.com	1.0	1.258
186	MIPA	Jam Istirahat	facebook.com , google.com	0.667	0.945
187	MIPA	Jam Istirahat	facebook.com , fbcnd.net	0.519	1.26
188	MIPA	Jam Istirahat	blogspot.com , blogger.com	1.0	3.39
189	MIPA	Jam Istirahat	blogger.com , blogspot.com	1.0	3.39
190	MIPA	Diluar Jam Kerja	youtube.com , google.com	1.0	2.11
191	MIPA	Diluar Jam Kerja	twitter.com , facebook.com	1.0	1.898
192	MIPA	Diluar Jam Kerja	twitter.com -> facebook.com	0.834	1.583
193	MIPA	Diluar Jam Kerja	mozilla.com , bbc.co.uk	0.715	2.709
194	MIPA	Diluar Jam Kerja	google.com , youtube.com	0.556	2.107
195	MIPA	Diluar Jam Kerja	google.com , facebook.com	0.556	1.056
196	MIPA	Diluar Jam Kerja	google.com -> facebook.com	0.667	1.266
197	MIPA	Diluar Jam Kerja	facebook.com , twitter.com	0.6	1.899
198	MIPA	Diluar Jam Kerja	facebook.com , google.com	0.5	1.055
199	MIPA	Diluar Jam Kerja	bbc.co.uk , mozilla.com	1.0	2.711
200	Teknik	Jam Kerja	twitter.com , google.com	0.648	2.455
201	Teknik	Jam Kerja	twitter.com -> google.com	0.648	2.455
202	Teknik	Jam Kerja	mozilla.com -> google.com	0.917	3.474
203	Teknik	Jam Kerja	google.com , twitter.com	0.551	2.46

204	Teknik	Jam Kerja	google.com , facebook.com	0.6	3.489
205	Teknik	Jam Kerja	google.com -> facebook.com	0.551	3.204
206	Teknik	Jam Kerja	facebook.com , google.com	0.924	3.5
207	Teknik	Jam Kerja	blogspot.com , blogger.com	0.917	5.804
208	Teknik	Jam Kerja	blogger.com , blogspot.com	0.917	5.804
209	Teknik	Jam Istirahat	google.com , facebook.com -> facebook.com	0.577	0.758
210	Teknik	Jam Istirahat	yahoo.id.com , google.com	0.889	1.037
211	Teknik	Jam Istirahat	google.com , facebook.com	0.723	0.949
212	Teknik	Jam Istirahat	google.com -> facebook.com	0.639	0.839
213	Teknik	Jam Istirahat	facebook.com , google.com	0.813	0.948
214	Teknik	Diluar Jam Kerja	mozilla.com -> facebook.com , bbc.co.uk	0.77	1.541
215	Teknik	Diluar Jam Kerja	youtube.com , gstatic.com	0.767	1.657
216	Teknik	Diluar Jam Kerja	youtube.com , google.com	0.867	1.141
217	Teknik	Diluar Jam Kerja	youtube.com -> google.com	0.667	0.878
218	Teknik	Diluar Jam Kerja	youtube.com , facebook.com	0.801	1.03
219	Teknik	Diluar Jam Kerja	youtube.com -> facebook.com	0.801	1.03
220	Teknik	Diluar Jam Kerja	twitter.com , google.com	0.852	1.122
221	Teknik	Diluar Jam Kerja	twitter.com , facebook.com	0.889	1.143
222	Teknik	Diluar Jam Kerja	twitter.com -> facebook.com	0.852	1.096
223	Teknik	Diluar Jam Kerja	mozilla.com -> google.com	0.639	0.841
224	Teknik	Diluar Jam Kerja	mozilla.com , facebook.com	0.556	0.715
225	Teknik	Diluar Jam Kerja	mozilla.com -> facebook.com	0.723	0.93
226	Teknik	Diluar Jam Kerja	mozilla.com , bbc.co.uk	0.75	1.5
227	Teknik	Diluar Jam Kerja	mozilla.com -> bbc.co.uk	0.556	1.113
228	Teknik	Diluar Jam Kerja	gstatic.com , youtube.com	0.921	1.657
229	Teknik	Diluar Jam Kerja	gstatic.com , google.com	0.84	1.106
230	Teknik	Diluar Jam Kerja	gstatic.com , facebook.com	0.801	1.03

231	Teknik	Diluar Jam Kerja	googlesyndication.com , google.com	0.84	1.106
232	Teknik	Diluar Jam Kerja	google.com , youtube.com	0.635	1.143
233	Teknik	Diluar Jam Kerja	google.com -> youtube.com	0.537	0.966
234	Teknik	Diluar Jam Kerja	google.com , twitter.com	0.561	1.123
235	Teknik	Diluar Jam Kerja	google.com -> mozilla.com	0.537	0.806
236	Teknik	Diluar Jam Kerja	google.com , gstatic.com	0.513	1.108
237	Teknik	Diluar Jam Kerja	google.com , googlesyndication.com	0.513	1.108
238	Teknik	Diluar Jam Kerja	google.com , facebook.com	0.781	1.004
239	Teknik	Diluar Jam Kerja	google.com -> facebook.com	0.708	0.911
240	Teknik	Diluar Jam Kerja	fbcdn.net , facebook.com	1.0	1.286
241	Teknik	Diluar Jam Kerja	facebook.com , youtube.com	0.572	1.029
242	Teknik	Diluar Jam Kerja	facebook.com , twitter.com	0.572	1.144
243	Teknik	Diluar Jam Kerja	facebook.com , google.com	0.762	1.003
244	Teknik	Diluar Jam Kerja	facebook.com , fbcdn.net	0.5	1.286
245	Teknik	Diluar Jam Kerja	bbc.co.uk , mozilla.com	1.0	1.5